RECONSTRUCTION OF CHROMOTOMOGRAPHIC IMAGING SYSTEM INFRARED HYPERSPECTRAL SCENES

THESIS

Malcolm G. Gould
Flight Lieutenant, Royal Australian Air Force

AFIT/GE/ENG/05-04

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views and conclusions contained in this thesis are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory, Department of Defense, United States Government, Royal Australian Air Force, Australian Department of Defence, or Government of the Commonwealth of Australia.

AFIT/GE/ENG/05-04

RECONSTRUCTION OF CHROMOTOMOGRAPHIC IMAGING SYSTEM INFRARED HYPERSPECTRAL SCENES

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Malcolm G. Gould, B.E. (Hons), Australian National University

Flight Lieutenant, Royal Australian Air Force

21 March 2005

AFIT/GE/ENG/05-04

RECONSTRUCTION OF CHROMOTOMOGRAPHIC IMAGING SYSTEM INFRARED HYPERSPECTRAL SCENES

THESIS

Malcolm G. Gould, B.E. (Hons), Australian National University

Flight Lieutenant, Royal Australian Air Force

Approved:

Stephen C. Cain　　17 Feb 2005

Dr. Stephen Cain　　Date
Assistant Professor of Electrical Engineering
Thesis Advisor

Matthew E. Goda　　18 FEB 2005

Maj. Matthew Goda　　Date
Assistant Professor of Electrical Engineering
Committee Member

18 Feb 2005

Dr. Glen Perram　　Date
Professor of Physics
Committee Member

Acknowledgements

Thankyou to all the people associated with the Royal Australian Air Force (RAAF) and United States Air Force for making the opportunity to study at AFIT a reality. It is no small feat to coordinate an international exchange program and the staff at RAAF Training Command, RAAF Overseas Administration Cell, the Australian Embassy in Washington D.C. and AFIT International Affairs have all done a great job.

A big thanks also to all the instructors and staff of both the Department of Electrical and Computer Engineering and the Department of Engineering Physics at AFIT. I have received exceptional instruction and support from a group of highly motivated people. I will fondly remember the professional ground covered at AFIT and the steepness of the learning curve.

I extend my sincerest appreciation to Dr. Cain for acting as my thesis advisor and providing great direction and support throughout both my thesis and stay at AFIT. A special thanks also to Major Goda and Dr. Perram for providing tireless support through my AFIT program and for being involved with this research. Thank-you also to my electrical engineering classmates; we've been challenged and have survived with senses of humor intact.

To my wife I express my deepest gratitude for your love, patience and inspiration. Your companionship, understanding and support has guided me through a busy couple of years. As life goes on, I look forward to sharing a bright future with you. Thanks also to the blonde blur, for taking our move all in her stride and always finding new ways to entertain.

Thankyou to our parents, sisters and brothers for providing some of the logistical support required when putting life on hold in one country while moving to another. Thanks also for providing the upbringing required for this opportunity to present itself in the first place, and also understanding our absence from the family for the past 20 months.

Finally, thankyou to all our local friends. We are so grateful to have met you and hope one day to catch up for a jam, a raft, a sausage roll and a tiramisu.

Malcolm G. Gould

Table of Contents

List of Figures

Figure Page

List of Tables

List of Symbols

Symbol		Page
f_{FL}	Focal length of focusing lens	17
r_λ	Radial spectral shift	17
θ_{DVP}	DVP rotation angle	18
$S(\theta_{DVP}, \lambda)$	Cartesian spectral shift function	19
x_s	Horizontal detector pixel shift	19
y_s	Vertical detector pixel shift	19
$\otimes$	2D-convolution	21
PSF_λ	Unshifted spectral PSF	22
PSF_θ	Detector Optics System PSF	23
$I(\theta_{DVP}, x_d, y_d)$	Image at detector (realization of the deterministic variable i)	25
$O(\lambda, u, v)$	Input object hyperspectral cube	25
$i(\lambda, \theta, x, y)$	Image at detector (random variable)	26
$PSF_\theta(\lambda, \theta, x, y)$	Detector Optics System PSF (explicit notation)	26
$d(\lambda, \theta, x, y)$	Data at detector (random variable)	27
$n(\lambda, \theta, x, y)$	Background noise (random variable)	27
(x_o, y_o)	Coordinates of one detector pixel	28
$D(\theta_o, x_o, y_o)$	Data at detector (realization of the random variable d)	28
Λ	Number of Spectral Bins	30
$\widehat{O}(\lambda, u, v)$	Estimate of input object hyperspectral cube	30
$\mathcal{F}$	Fourier Transform	31
$\mathcal{F}^{-1}$	Inverse Fourier Transform	32
$d_x(\lambda, \theta, x)$	Horizontal vector sum of data at detector (random variable)	35
$i_x(\lambda, \theta, x)$	Horizontal vector sum of image at detector (deterministic variable)	35
$n_x(\lambda, \theta, x)$	Horizontal vector sum of noise at detector (random variable)	35
$D_x(\theta_o, x_o)$	Horizontal vector sum of detector (realization of the random variable d_x)	35
$t_{atm}(\lambda)$	Atmospheric transmission coefficient	39
$\widehat{O}_{atm}(\lambda, u, v)$	Estimate of image object hyperspectral cube (attenuated)	40
Υ	Sum of Atmospheric Transmission Coefficients	43
Φ_p	Photon Flux	45
Δt	Detector integration time	45

List of Abbreviations

Abbreviation		Page
3D	Three-Dimensional	1
CTIS	Chromotomographic Imaging System	4
FOV	Field of View	6
MSI	Multiplexing Spectral Imager	7
2D	Two-Dimensional	7
AFIT	Air Force Institute of Technology	8
PSF	Point Spread Function	8
FOS	Forward Optics System	8
POS	Prism Optics System	8
DVP	Direct Vision Prism	8
LiF	Lithium Fluoride	8
BaF_2	Barium Fluoride	8
DOS	Detector Optics System	9
FPA	Focal Plane Array	9
SVD	Singular Value Decomposition	10
PCA	Principal Components Analysis	10
POCS	Projection Onto Convex Sets	10
OTF	Optical Transfer Function	11
f/#	f/number	14
LSI	Linear Shift Invariant	21
PMF	Probability Mass Function	27
iid	Independent and Identically Distributed	27
1D	One-Dimensional	33
IR	Infrared	33
4D	Four-Dimensional	38
RHS	Right-Hand-Side	38
REM	Reconstruction Error Metric	73

AFIT/GE/ENG/05-04

Abstract

Hyperspectral imagery providing both spatial and spectral information has diverse applications in remote sensing and scientific imaging scenarios. The development of the Chromotomographic Imaging System (CTIS) allows simultaneous collection of both spatial and spectral data by a two-dimensional (2D) focal plane detector array. Post-processing of the 2D detector data reconstructs the three-dimensional (3D) hyperspectral content of the imaged scene.

This thesis develops Estimation Theory based algorithms for reconstructing the hyperspectral scene data. The initial algorithm developed reconstructs the 3D hyperspectral scene data cube. An additional algorithm reconstructs a matrix comprised of one spectral dimension and one compound spatial dimension. This spatial dimension consists of a vector sum along one spatial dimension of the 3D hyperspectral data cube. Methods for including the effects of atmospheric attenuation on the light over the propagation path are also included.

The algorithms are evaluated using test cases consisting of blackbody point sources, monochromatic extended sources and blackbody extended sources. The results show good performance for reconstructing the absolute radiometry and spatial features of a hyperspectral scene data cube. Reconstructed temperatures are within 4.1% of the original temperature for 2D reconstructions and within 6.0% for vector reconstructions. These algorithms also do not significantly degrade in the presence of noisy detector data. The radiometrically accurate reconstruction of atmospherically attenuated detector data proves viable for wavelengths where there is sufficient photon levels at the detector. Upon further processing reconstructed temperatures are within 3.6% of the original temperature for 2D reconstructions and within 5.8% for vector reconstructions.

The vector algorithm also exhibits stable performance behaviour when reconstructing a temporally evolving hyperspectral scene data cube. The resulting reconstructed temperature of a data cube pixel is within 2.3% of the average temperature at the original scene location.

RECONSTRUCTION OF CHROMOTOMOGRAPHIC IMAGING SYSTEM INFRARED HYPERSPECTRAL SCENES

I. Introduction

This chapter describes the concept of hyperspectral imagery and the format of the associated hyperspectral data cubes. Hyperspectral imagery has applications in diverse imaging scenarios providing substantial motivation to conduct further research in this area. The specific research goals of this thesis are stated in this chapter along with an overview of the organization of this document.

1.1 Hyperspectral Imagery

Hyperspectral imagery can be envisaged as a three-dimensional (3D) data cube consisting of two spatial dimensions and one spectral dimension. This is further visualized by perceiving a series of monochromatic images of the same scene stacked in the data cube. The sum of the data cube along the spectral, or wavelength, dimension results in a broadband image of the scene within the measured wavelengths as demonstrated in Figure 1.1. This is similar to a common photograph if the data cube is collected in the visual spectral range. The spectral composition of a spatial location in the scene may also be examined by viewing the location along its spectral dimension. With the application of radiometry the temperature of the spatial location can then be determined. Collecting multiple data cubes of a scene changing over time adds a fourth dimension to the data set and the series of data cubes may be viewed as a hyperspectral movie.

The convention presented in the literature regards monochromatic, or panchromatic, imagery as an image collected at one wavelength. Multispectral imagery is considered as imagery collected at a number of wavelengths, but not normally more than about a dozen. Hyperspectral imagery is classed as imagery collected at any more than a dozen wavelengths. This can range from tens to hundreds of wavelengths in currently fielded systems.

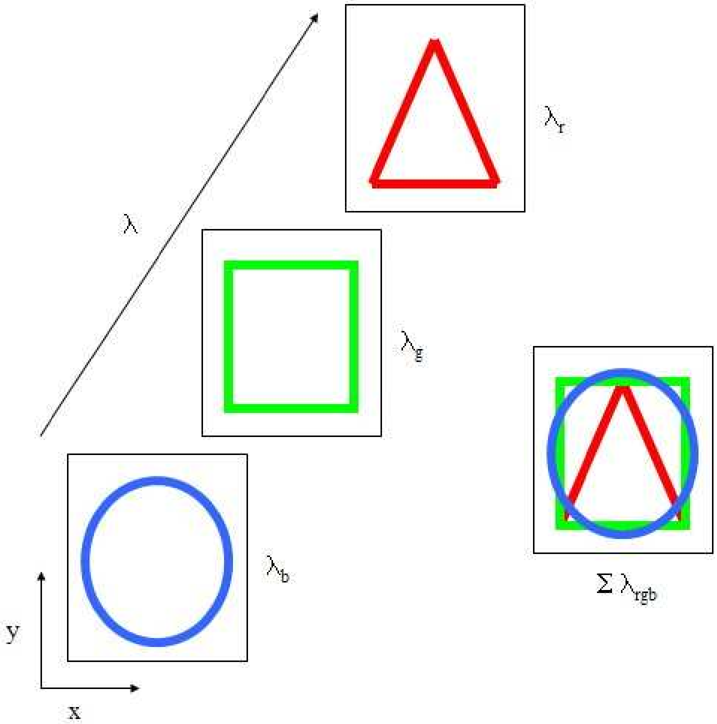


Figure 1.1: This figure shows three monochromatic images (red, green and blue) within a multispectral data cube. The data cube is composed of two spatial dimensions, x and y, and one spectral dimension, λ. The sum of the images along the spectral dimension of the data cube results in a broadband image of the red, green and blue wavelengths.

1.2 Research Motivation

Objects contained in an imaged scene will absorb, emit and reflect electromagnetic radiation. The radiating properties of the objects directly correlate to their material properties. Hence analysis of hyperspectral scene data can be used to discriminate objects contained in the imaged scene.

It is interesting to note the diverse application of hyperspectral imaging to existing and potential fields drawn from both civilian and military applications as follows [19] [9]:

- Environmental Monitoring
  - Pollutant detection in drainage systems
  - Emergency response and plume tracking
  - Crop health monitoring
  - Mineral deposit surveying
- Military Applications
  - Camouflage detection
  - Landmine detection
  - Battlefield monitoring of chemical and biological agents
- Law Enforcement
  - Counterfeit currency detection
  - Marijuana detection in natural vegetation
  - Detection of illicit drug manufacturing by-products
  - Target detection in Search and Rescue operations

- Medical Applications
    - Optical biopsy, eg, cancer cell detection
    - Functional mapping of brain
- Manufacturing / Industrial Applications
    - Weld quality inspection
    - Rust detection
    - Detection of surface defects in thin films

Hence the extension of the hyperspectral knowledge base is potentially applicable to a wide range of applications in different professional disciplines.

This thesis specifically addresses the issue of post-processing hyperspectral imagery of both static and time varying scenes. This potentially allows further applications for the collection and processing of hyperspectral scene data in Remote Sensing roles and as a recording sensor for engineering Test and Evaluation.

1.3 Research Goals

This thesis focuses on developing and evaluating reconstruction algorithms suitable for post-processing the detector images generated by the Chromotomographic Imaging System (CTIS) developed by Mooney *et al* [15]. This research also complements previous work undertaken by AFIT which has the end goal of developing a CTIS instrument. Specific areas addressed in this thesis include:

- The development and evaluation of a CTIS detector reconstruction algorithm which maintains absolute radiometric accuracy.
- The development and evaluation of a CTIS detector reconstruction algorithm capable of imaging a temporally evolving input image scene.

- The investigation of two-dimensional versus one-dimensional CTIS detector reconstruction algorithms in terms of absolute radiometric performance.
- The development and evaluation of a CTIS detector reconstruction algorithm which can be applied to atmospherically attenuated detector data.
- The evaluation of the CTIS detector reconstruction algorithms in the presence of photon noise at the detector.

1.4 Organization

Chapter II introduces current hyperspectral data collection methods and provides an overview of the CTIS modelled in this thesis. Chapter III develops the discrete system model of the optical components of the CTIS and derives the reconstruction algorithm required to estimate the input hyperspectral data cube. Chapter IV presents the test case imaging scenarios used to evaluate the performance of the reconstruction algorithms. Chapter V presents the test case results and provides an analysis of the performance of the reconstruction algorithms. Chapter VI concludes this thesis providing key results and recommendations for further research.

II. Hyperspectral Data Collection

This chapter discusses conventional hyperspectral collection techniques and their inherent disadvantages. The concept and development of the Chromotomographic Imaging System (CTIS) is presented. The CTIS optical configuration and post-processing requirements are documented along with its concept of operation for collecting hyperspectral data.

2.1 Hyperspectral Data Collection Methods

As discussed in section 1.1 hyperspectral data cubes consist of two spatial dimensions and one spectral dimension. There are several conventional methods for collecting hyperspectral information which fundamentally entail building the data cube along either the spatial or spectral dimensions. For example, one method scans a slit across the instrument field of view (FOV). The diffraction induced by the slit spectrally disperses the light providing spectral data for the spatial location of the slit. Another method is to determine spectral content based on wave interference generated by a Michelson interferometer. A third method is to use spectral filters to capture monochromatic images of the scene at different wavelengths. Hence this method constructs the data cube by stacking the spatial images along the spectral dimension.

These methods all suffer from low optical throughput as the hyperspectral data is collected piece-wise. Other disadvantages include the inability to collect complete data on temporally changing scenes and high vibration sensitivity particulary applicable to interference based instruments. Further information on conventional hyperspectral collection systems is found in chapter 1 of [6], chapter 2 of [11] and chapter 2 of [1].

2.2 Chromotomographic Imaging System Overview

The hyperspectral system modelled in this thesis is known as the Chromotomographic Imaging System (CTIS) and is founded on work by Mooney, Brodzik and An [2,3,15,16]. The configuration of the CTIS consists of a Forward Optics System, Prism Optics System and Detector Optics System as shown in Figure 2.1. The CTIS is also referred to as a Multiplexing Spectral Imager (MSI) due to its operating principal of concurrently collecting both spatial and spectral information onto a two-dimensional (2D) detector.

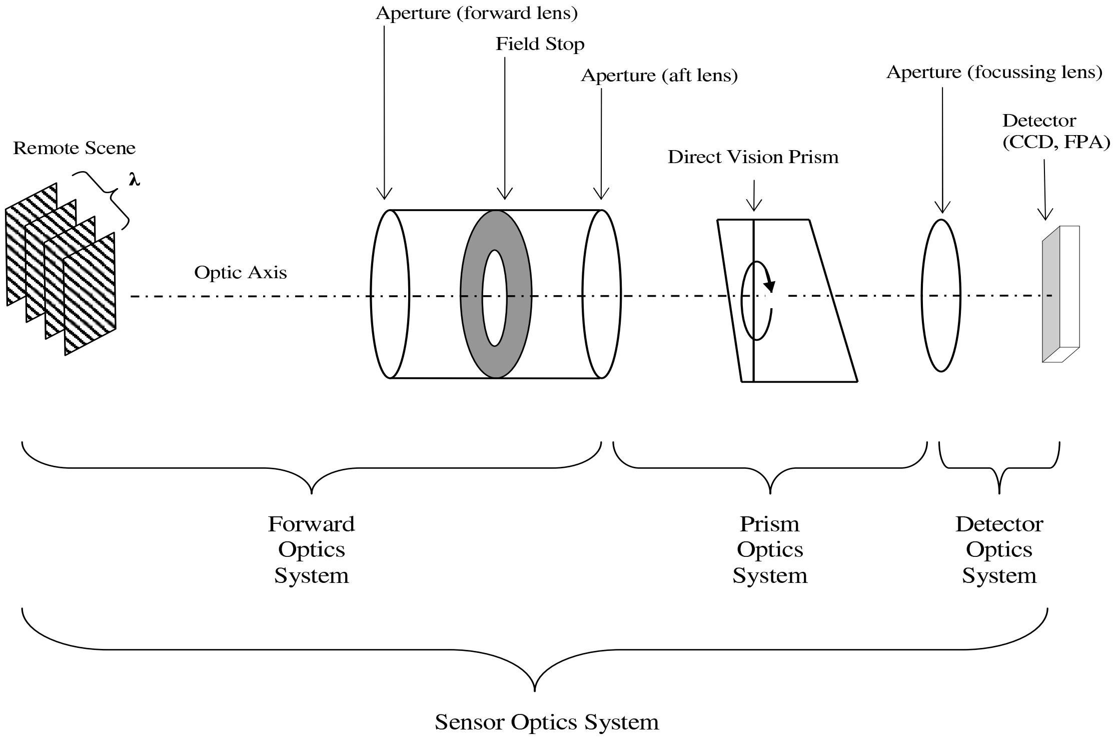


Figure 2.1: This figure shows the sub-systems which comprise the CTIS. These include a Forward Optics System (FOS), Prism Optics System (POS) and a Detector Optics System (DOS). [6]

The main advantage of the CTIS is a high data throughput via the optical components and subsequent efficient use of the collected radiation. This effectively allows the CTIS to be regarded as a staring hyperspectral sensor with a potential application for sensing transient events in an object scene. However due to its multiplexing nature, the CTIS also requires a complex algorithm to reconstruct the 3D hyperspectral cube of the object scene from the 2D image at the detector.

The United States Air Force Institute of Technology (AFIT) has conducted previous work on the CTIS. Work conducted by Dearinger [6] developed a high fidelity Matlab® optical propagation model of the CTIS in order to determine the spectral point spread functions (PSF) for a given optical component configuration. Work conducted by Gustke [11] examined and developed reconstruction algorithms based on the application of matrix inversion methods. Work conducted by LeMaster in [13] verified the optical propagation models with laboratory equipment physically configured as a CTIS.

Further information on the CTIS is found in chapter 2 of [6] by Dearinger with the fundamental sub-systems discussed as follows:

2.2.1 Forward Optics System. The Forward Optics System (FOS) senses hyperspectral scene data and consists of a forward lens, cold field stop and an aft lens. The FOS serves to limit the spatial extent of the scene data and performs collimation of the received light.

2.2.2 Prism Optics System. The Prism Optics System (POS) performs spectral dispersion of the hyperspectral scene data using a Direct Vision Prism (DVP) with the resulting effect demonstrated in Figure 2.2. The DVP consist of two back-to-back prisms made from different optical materials. The front prism half is made from Lithium Fluoride (LiF) while the rear prism half is made from Barium Fluoride (BaF_2). Spectral dispersion is attributed to the wavelength dependent refractive index of the optical material and the angles of the prism entrance and exit faces. The spectral refractive indices for both LiF and BaF_2 are shown in appendix A. The counteracting geometric and refractive properties of the prism halves in the DVP allow it to be tuned to both a specific undeviated wavelength and maximum refractive angles resulting from the minimum and

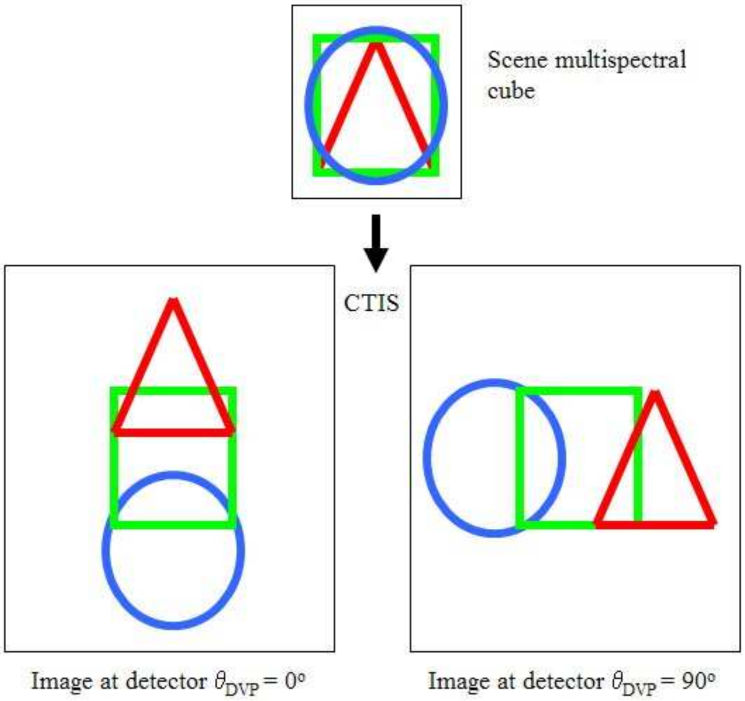

Scene multispectral cube
CTIS
Image at detector $\theta_{DVP} = 0°$
Image at detector $\theta_{DVP} = 90°$

Figure 2.2: This figure shows a simple example of the spectral dispersion effect induced by the Direct Vision Prism (DVP). The monochromatic images in this multispectral scene are spatially dispersed by the wavelength dependent refraction within the DVP resulting in a spatial separation at the detector. As the DVP rotates detector images at different DVP rotation angles are collected, with two examples shown in the figure. The DVP is also optically configured to allow light from one wavelength to pass undeviated to the detector. In the case of the figure, light at a green wavelength is undeviated.

maximum detectable wavelengths. The final feature of the DVP is that it rotates around the optic axis. This enables different views of the hyperspectral scene data to be obtained at the detector.

2.2.3 Detector Optics System. The Detector Optics System (DOS) consists of a focusing lens and focal plane array (FPA) as the detector device. Light from the DVP contains wavelength and DVP rotation angle dependent tilts. The focusing lens focuses this tilted light at different shifted positions on the detector. Consequently the shifts in the detector are also wavelength and DVP rotation angle dependent.

2.2.4 Reconstruction of Scene Hyperspectral Image. A CTIS detector image requires post-processing by a reconstruction algorithm to estimate the corresponding input data cube. The reconstruction algorithm uses multiple 2D detector images and the overall CTIS transfer function to estimate the original 3D hyperspectral data cube.

As described by Gustke in [11] the state-of-the-art CTIS reconstruction algorithms rely on linear systems principles. This requires inverting the CTIS transfer function in order to determine the input data cube from a given CTIS detector image. Hence the need (as a minimum) to collect detector image at the same number of DVP rotation as wavelength bins. However the resulting matrix describing the transfer function is singular as the detector data collected at different DVP rotation angles is not independent. This leads to the use of Singular Value Decomposition (SVD) as described by Strang in appendix A of [18] to determine the pseudo-inverse of transfer function matrix.

Subsequent reconstruction of the input data cube using the pseudo-inverse will contain missing hyperspectral data. Gustke conducts a trade-off analysis of reconstruction performance using several methods to approximate the missing data including Principal Components Analysis (PCA), Projection Onto Convex Sets (POCS) and Non-Iterative PCA. Modifications to the algorithms are also implemented providing additional constraints to reconstructed data. Conclusions indicate a requirement to improve the absolute radiometric accuracy of the calculated input hyperspectral data cube generated by these reconstruction methods.

III. System Modelling Methodology

This chapter develops the discrete model of the CTIS. It includes the modelling assumptions and design parameters used to implement the model. An optical propagation model of both the POS and DOS are developed to simulate CTIS-like detector images. The 2D and vector input hyperspectral reconstruction algorithms are derived along with methods for including the effects of atmospheric attenuation on the detector images.

3.1 Discrete Modelling of the Chromotomographic Imaging System

Discrete modelling of the CTIS in Matlab® uses elements of geometric optics, Zernike phase screen generation, Rayleigh-Sommerfeld diffraction theory and Fourier optics. The model uses geometric optics to determine the wavelength dependent shifts. Zernike phase screens are used to simulate the rotation of the DVP. Rayleigh-Sommerfeld diffraction theory is used to calculate the the unshifted, wavelength dependent PSFs. Fourier optics is used to determine the optical transfer function (OTF) of the modelled components.

3.1.1 Model Assumptions. The goal of implementing a CTIS system model is to produce CTIS-like images at the detector in order to evaluate reconstruction algorithms. These images will exhibit the wavelength and DVP rotation angle dependent shifts discussed in section 2.2.3. The CTIS will also induce a wavelength dependent diffractive spreading of the light. Hence the monochromatic images in the hyperspectral cube will each undergo some blurring due to the CTIS optical components. However this blurring is characterized by the spectral PSF of the system which is the response of the system to a point source located at infinite range. Thus in order to concentrate on the reconstruction algorithms, the methodology used in this thesis makes several simplifying assumptions for the generation of detector data.

The assumptions used in modelling the CTIS include:

3.1.1.1 Light entering the DVP is collimated. The purpose of the FOS is to collimate the collected light and limit the spatial extent of the image. These are actually conflicting functions as the field stop will produce some diffractive effects resulting in the light not being completely collimated. The diffractive effects are characterized by dimensions of the field stop aperture which limit the spatial frequency of the collected image. This can be viewed as part of the overall PSF and can be determined by modelling the light propagation in the FOS as demonstrated by Dearinger in [6]. An actual CTIS will also have to undergo complete PSF characterization in order to implement an applicable reconstruction algorithm.

The CTIS model in this thesis limits the spatial extent of the scene by constructing hyperspectral cubes of a known dimension. This CTIS model also assumes that the light entering the DVP is collimated. Consequently the FOS is not modelled in this thesis with the exception of using the forward lens diameter in radiometric calculations.

3.1.1.2 The DVP is a "thin" optical medium. A thin optical component is where a light ray enters and exits the component at approximately the same lateral coordinates with respect to the optical axis as defined by Goodman in chapter 5 of [10]. Given the dimensional and material characteristics of the DVP it is unlikely that it exhibits thin optical performance. However the overall effect of the DVP is to produce wavelength and DVP rotation angle dependent tilt on the exiting wavefront which produces a shift in the PSF at the detector. Again the degree of tilt and the resulting shift at the detector is dependent on the PSF of the DVP.

The CTIS model in this thesis uses geometric optics to determine the location of the shift at the detector. The calculation as detailed in section 3.2 is based on the geometry of the DVP and the translational distance between the DVP and focusing lens. The unshifted spectral PSF (PSF_λ)

of the modelled components is then convolved with the respective shift location (equivalent to 2D Dirac δ function) to produce the overall CTIS PSF.

3.1.1.3 Diffraction of Focusing Lens over-emphasized. The simulation is modified to over-emphasize the diffractive effects as the actual system with parameters defined in table 3.1 produces an Airy pattern within one pixel of the detector. This is regarded as good fortune for constructing a real system as the Airy pattern is effectively sampled as a delta function at the detector. However in order to provide more challenge to the reconstruction algorithms developed in section 3.4 it is desirable to broaden each PSF_λ over multiple pixels. A real optical system will also impart abberations which provide some shape to the PSF. The PSF is broadened by firstly setting the pixel pitch to $3\mu m$ when performing the diffraction calculation, and then treating each pixel as the larger actual detector pitches shown in table 3.1 ($66.67\mu m$ and $100\mu m$) when convolving PSF_λ with the respective shift location.

3.1.2 Model Design Parameters. The design parameters used in the model are shown in Table 3.1. Of note in the table is that all the lenses have the same f-number (f/#) which maintains conservation of energy of the collected light between the model sub-systems. The selection of model parameters follows the work conducted by Dearinger in [6] and is again aimed at producing a CTIS system model for the generation of hyperspectral data cubes in order to evaluate reconstruction algorithms. The development of the model proceeds using 15 spectral bins with each bin $0.2\mu m$ wide and having bin centers as shown in Table 3.2.

Table 3.1: CTIS Model Design Parameters

Component	**Design Parameter**	**Value**
Object Scene	Wavelength range	2.0 to $5.0\mu m$
Forward lens	Diameter	$0.1m$
	Focal length	$1.0m$
Aft lens	Diameter	$0.022m$
	Focal length	$0.22m$
Focusing Lens	Diameter	$0.05m$
	Focal length(f_{FL})	$0.5m$
Direct Vision Prism	Undeviated Wavelength	3.6 μm
	Front angle	30.00^o
	Middle angle	0^o
	Aft angle	23.95^o
LiF	$n_3(2\mu m)$	1.37875
	$n_3(5\mu m)$	1.32661
BaF_2	$n_4(1.97\mu m)$	1.46470
	$n_4(5.14\mu m)$	1.45014
Detector	Array size	256 × 256
	Pixel pitch(A)	$100\mu m$
	Pixel pitch(B)	$66.67\mu m$
Reconstruction	Spectral bins	15
	Spectral Resolution	$\frac{3}{15}\mu m$

Table 3.2: Wavelength Bin Centers where each bin is $0.2\mu m$ wide.

Wavelength Bin	**Bin Center**	**Wavelength Bin**	**Bin Center**
	(μm)		(μm)
1	2.1	9	3.7
2	2.3	10	3.9
3	2.5	11	4.1
4	2.7	12	4.3
5	2.9	13	4.5
6	3.1	14	4.7
7	3.3	15	4.9
8	3.5		

3.2 Prism Optics System Modelling

The spectral dispersion of the DVP is dependent on both the optical properties of the component materials and the physical geometry of the DVP.

3.2.1 Spectral Refractive Indices of the DVP Materials. The spectral refractive indices in the applicable wavelength range for both LiF and BaF_2 are shown in appendix A. These refractive indices are linearly interpolated to find the spectral refractive index at the center of each wavelength bin as shown in Figure 3.1.

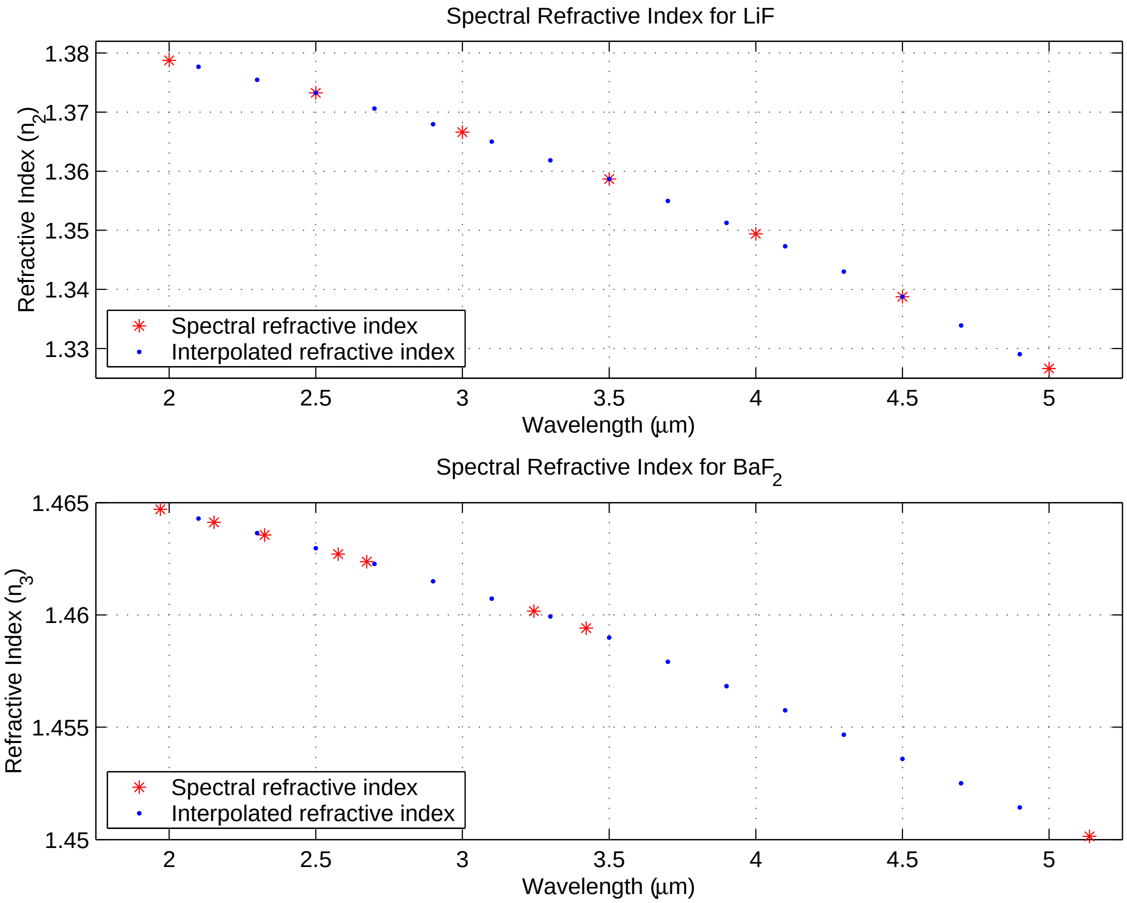


Figure 3.1: This figure shows the linear interpolation of the spectral refractive indices for both LiF and BaF_2 at the center of each wavelength bin.

3.2.2 Physical Geometry of the DVP. The physical geometry of the DVP is as shown in Figure 3.2. The front prism half of the DVP is constructed from LiF with a spectral refractive index of $n_2(\lambda)$ and an entrance angle of $\alpha_1 = 30^o$. The rear prism half of the DVP is constructed from BaF$_2$ with a spectral refractive index of $n_3(\lambda)$ and an exit angle of $\alpha_3 = 23.95^o$. The angle of the interface between the prism halves is $\alpha_2 = 0^o$. The refractive index of air is modelled as a constant with $n_1 = 1$. This DVP geometry allows light at a wavelength of $\lambda = 3.6\mu m$ to pass through the prism undeviated.

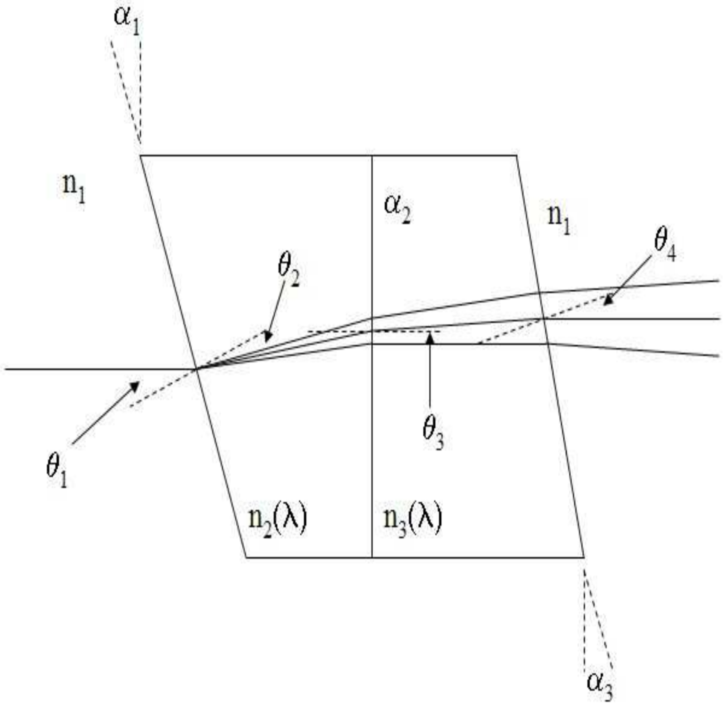


Figure 3.2: This figure shows the spectral dispersion effects of the Direction Vision Prism (DVP) and labels the angles used.

The spectral dispersion of the DVP is modelled by the application of Snell's Law to the three surface interfaces. As stated by Hecht in chapter 4 of [12], Snell's Law is

$$n_i \sin\theta_i = n_t \sin\theta_t \tag{3.1}$$

where n_i and n_t are the refractive indices of the incident and transmitted mediums, and θ_i and θ_t are the incident and transmitted ray angles respectively. Observing that $\theta_1 = \alpha_1$, the following expressions can then be derived for the DVP ray angles

$$\theta_2 = \sin^{-1}\left(\frac{n_1}{n_2(\lambda)}\sin(\theta_1)\right) - \alpha_1 \tag{3.2}$$

$$\theta_3 = \sin^{-1}\left(\frac{n_2(\lambda)}{n_3(\lambda)}\sin(\theta_2)\right) + \alpha_3 \tag{3.3}$$

$$\theta_4 = \sin^{-1}\left(\frac{n_3(\lambda)}{n_1}\sin(\theta_3)\right) - \alpha_3 \tag{3.4}$$

where the α_1 term in equation 3.2 and α_3 terms in equations 3.3 and 3.4 are applied to maintain a constant vertical frame of reference. Knowledge of the exit ray angle, θ_4, in conjunction with the focal length, f_{FL}, of the Focusing Lens can be used to determine the radial shift at the detector for each wavelength by

$$r_\lambda = -f_{FL}\tan\theta_4 \tag{3.5}$$

where r_λ is the radial spectral shift and the negative sign accounts for the image inversion produced by the lens. Using equations 3.4 and 3.5 for the case of 15 wavelength bins and the interpolated spectral refractive indices from section 3.2.1, the radial spectral shifts resulting from the DVP and Focusing Lens are shown in Figure 3.3.

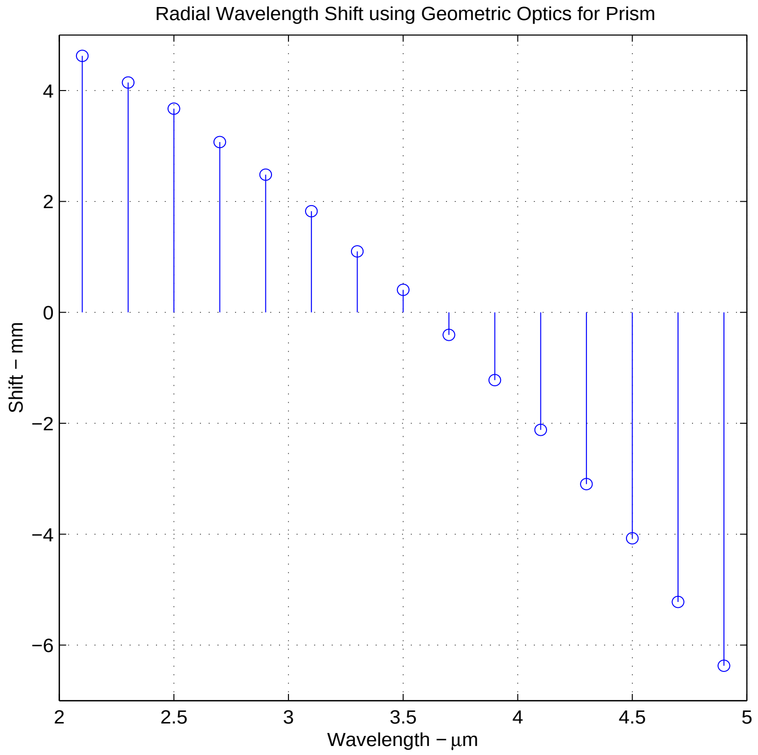


Figure 3.3: This figure shows the radial spectral shifts (r_λ) at the detector resulting from the Direct Vision Prism and Focusing Lens for 15 spectral bins. Note that $\lambda = 3.6\mu m$ is the undeviated wavelength and that the separation between consecutive shifts increases as wavelength increases.

The rotation of the DVP is implemented in the model using a multiple 2D tilt Zernike phase screen concept. However it is not required to implement the entire phase screen, only the geometric effect of the prism on the light rays at each rotation angle. This geometric effect consists of converting the radial spectral shift into a 2D cartesian spectral shift and is achieved using basic trigonometry as shown in Figure 3.4 and given by

$$x_s = r_\lambda \sin(\theta_{DVP}) \tag{3.6}$$

$$y_s = r_\lambda \cos(\theta_{DVP}) \tag{3.7}$$

where r_λ is the radial spectral shift, θ_{DVP} is the DVP clockwise rotation angle from vertical and (x_s, y_s) are the respective cartesian shifts in x (horizontal) and y (vertical) at the detector.

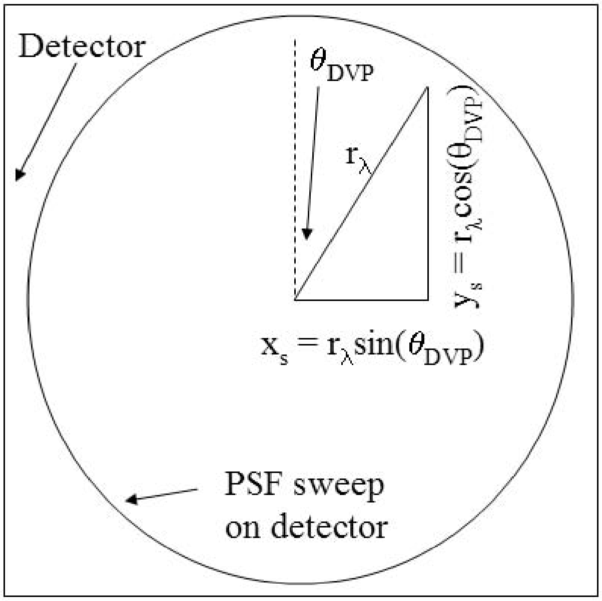

Detector
θ_{DVP}
r_λ
$\mathrm{y_s = r_\lambda cos(\theta_{DVP})}$
$\mathrm{x_s = r_\lambda sin(\theta_{DVP})}$
PSF sweep
on detector

Figure 3.4: This figure shows the derivation of cartesian spectral shifts at the detector which is dependent on the Direct Vision Prism and Focusing Lens.

By combining the DVP spectral dispersion, the DVP rotation angle and the focal length of the focusing lens, the cartesian spectral shifts at the detector as a function of DVP rotation angle can be obtained. The cartesian spectral shift function, $S(\theta_{DVP}, \lambda)$, is represented as

$$[x_s \; y_s] = S(\theta_{DVP}, \lambda) \tag{3.8}$$

where $[x_s, y_s]$ are the pixel coordinates of the shift at the detector due to an input DVP rotation angle, θ_{DVP}, and input spectral bin wavelength, λ. A closed form expression for $S(\theta_{DVP}, \lambda)$ is not derived but simulation results are shown in Figure 3.5 for 15 DVP rotation angles. Two detector pixel pitches are used for subsequent simulations depending on the imaging application as discussed in section 3.3.

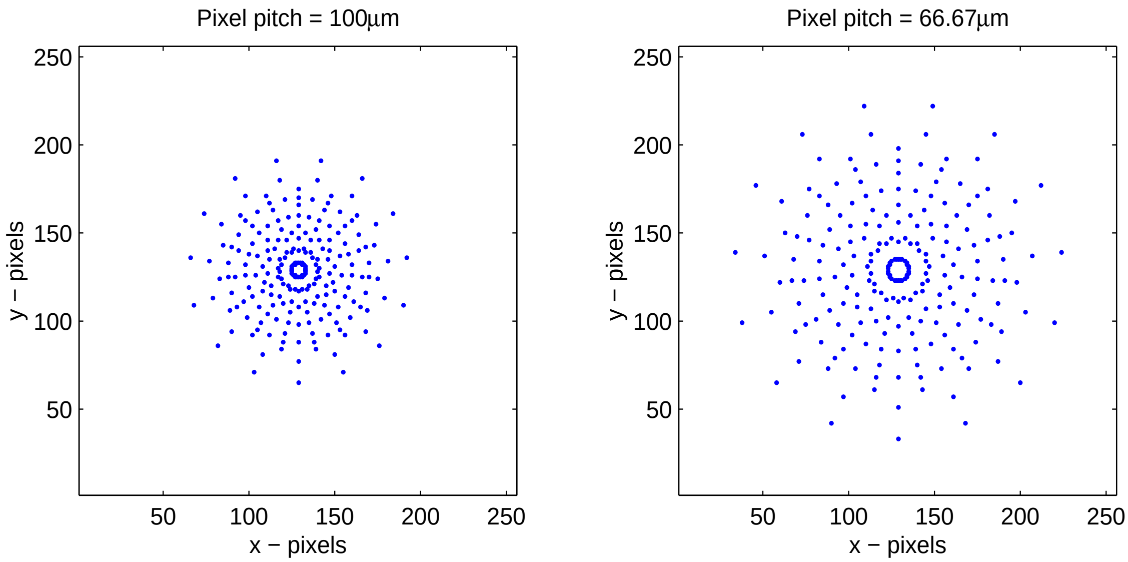


Figure 3.5: This figure shows cartesian spectral shifts at the two detector pitches, $100\mu m$ and $66.67\mu m$ resulting from the spectral dispersion and rotation of Direct Vision Prism and the focal length of the Focusing Lens. Note that this is a compound figure and the actual shifts will only appear at the corresponding rotation angle of the DVP. Hence this figure can be thought of as a sum of S over each rotation angle, ie, $\sum_{i_\theta=1}^{15} S(\theta_{DVP}, \lambda)$, where i_θ indexes the DVP rotation angles.

A final note on modelling the DVP is that the number of DVP rotation angles is purposely set equal to the number of spectral bins. This approach follows on from previous work conducted by Gustke in [11] and Mooney *et al* in [2, 3, 15, 16] in which reconstruction of the hyperspectral cube from the detector signal is implemented using linear algebra techniques. Hence the minimum number of spectral DVP rotation angles required is equal to the number of spectral bins in order to attempt a matrix inversion method. Further work is required to investigate the effects of an unequal number of spectral bins and DVP rotation angle on the reconstruction algorithms that are documented in section 3.4.

3.3 Detector Optics System Modelling

Geometric optics can only be used so far when modelling an optical system. Hence Rayleigh-Sommerfeld diffraction theory is used to model the diffraction effects of the Focusing Lens. The first step in this approach is to calculate the unshifted PSFs for each wavelength corresponding to each spectral bin. Modelling the DOS as a Linear Shift Invariant (LSI) system, the PSF for each wavelength will be shift invariant and thus independent of the tilt it has acquired from the DVP. Hence the system PSF for each DVP rotation angle can be "constructed" by combining DVP/Focusing Lens shift data and the spectral PSFs of the DOS through a 2D-convolution ($\otimes$) operation.

The Rayleigh-Sommerfeld diffraction integral, as presented by Goodman in [10], describes the wave optics propagation of light from between two parallel planes. This is mathematically expressed as

$$U(x_1, y_1) = \frac{1}{j\lambda} \iint\limits_{\Sigma} \frac{U(x_0, y_0)}{r_{01}} \exp^{jkr_{01}} \cos\theta \; dx_0 dy_0 \tag{3.9}$$

where $U(x_0, y_0)$ is the electromagnetic field over a transmitting aperture plane $\sum$ and $U(x_1, y_1)$ is the field at a receiving plane. The distance between points on the planes is denoted r_{01} and is a Pythagorean function of the perpendicular distance between the planes and transverse plane coordinates (x_0, y_0) and (x_1, y_1). The angle of r_{01} with respect to the propagation normal direction is denoted as θ. The integral is applicable to monochromatic light with wavelength λ and corresponding wave number $k = \frac{2\pi}{\lambda}$.

When cast in the discrete form applicable for wave optics modelling of propagation through a lens, the Rayleigh-Sommerfeld diffraction integral forms the sum as

$$U(x_d, y_d) \approx \frac{f\Delta^2}{j\lambda} \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{U(x_n, y_m) t_l(x_n, y_m)\ e^{\frac{j2\pi}{\lambda}\sqrt{f^2+(x_n-x_d)^2+(y_m-y_d)^2}}}{f^2+(x_n-x_d)^2+(y_m-y_d)^2} sinc\left(\frac{\pi\Delta x_d}{f\lambda}\right) sinc\left(\frac{\pi\Delta y_d}{f\lambda}\right) \tag{3.10}$$

where f is the lens focal length, λ is the wavelength of the light, n indexes N lens samples in the horizontal direction and m indexes M lens samples in the vertical direction. The field entering the lens is denoted $U(x_n, y_m)$ where (x_n, y_m) are the coordinates of discrete square samples of the lens with a side-length of Δ. The lens transformation is $t_l(x_n, y_m)$. The field at a single pixel on the detector is denoted $U(x_d, y_d)$ and is indexed to determine of the overall sampled field at the detector. The derivation of the above discrete Rayleigh-Sommerfeld diffraction sum for propagation from a lens to a detector is shown in appendix B.

For an optical system the PSF is defined as the system response to a distant point source. A point source located an infinite distance from the system will result in a plane wave at the receiving aperture. As the input at the focusing lens is a plane wave there is no phase difference across this aperture so it is modelled as an 11×11 matrix, ie, the small number of samples sufficiently samples the phase across the aperture. The circular shape of the lens aperture is implemented by the use of a masking function. As previously documented, the focal length of the Focusing Lens is $f_{FL} = 0.5m$.

The unshifted spectral PSFs are denoted PSF_λ where the λ subscript indicates wavelength dependence. Each PSF_λ is calculated on a 21×21 matrix which provides sufficient spatial extent for observing the resultant Airy pattern of the PSF. Note that each PSF_λ represents the intensity of the respective electromagnetic fields and is in units of $\frac{photons}{sec}$. As discussed in section 3.1.1.3 the simulation is modified to over-emphasize the diffractive effects. This is achieved by setting the pixel pitch to $3\mu m$ when performing the diffraction calculation and then treating each pixel as the larger actual detector pitches shown in Table 3.1 ($66.67\mu m$ and $100\mu m$). Note that as shown in

Figure 3.6, PSF_λ becomes broader and decreases in peak intensity with increasing wavelength. To maintain conservation of energy, the sum of each PSF_λ over the 21 by 21 matrix is normalized to sum to one.

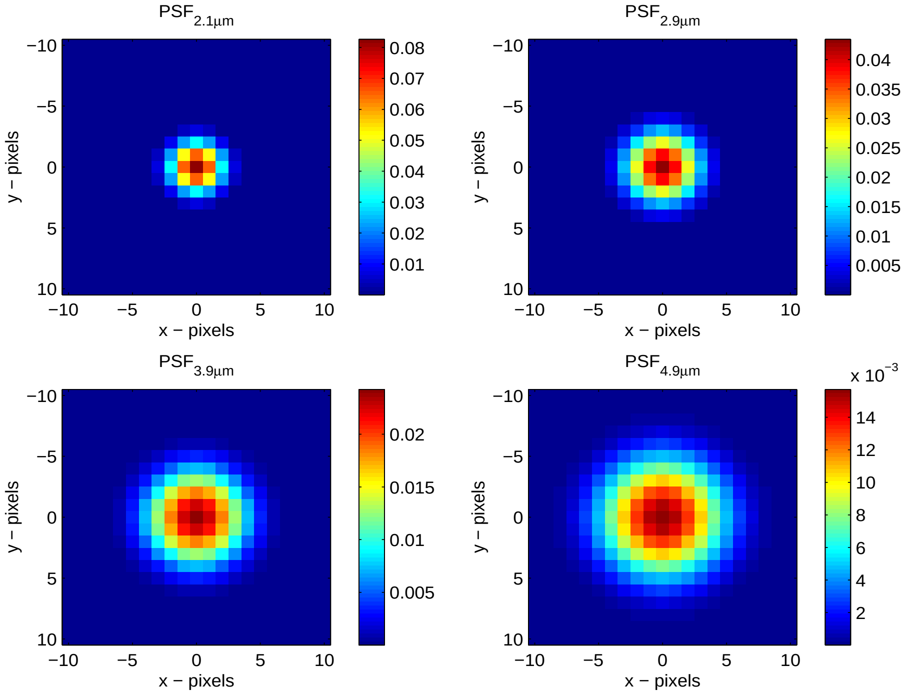


Figure 3.6: This figure shows the different unshifted spectral PSF_λ for $\lambda = 2, 2.9, 3.9$ and $4.9\mu m$. Each unshifted spectral PSF_λ sums to one over the 21×21 matrix.

The PSF of the DOS, PSF_θ, at each of the DVP rotation angles will be the sum of the unshifted PSF_λ located at their appropriate geometric shift shown in Figure 3.5. This is mathematically expressed as

$$PSF_\theta = \sum_{i_\lambda=1}^{15} PSF_\lambda \otimes S(\theta_{DVP}, \lambda) \qquad (3.11)$$

where i_λ indexes each wavelength bin. The PSF at the detector, $PSF_{\theta=0^o}$, for the vertical DVP rotation angle, $\theta_{DVP} = 0^o$ is shown in Figure 3.7.

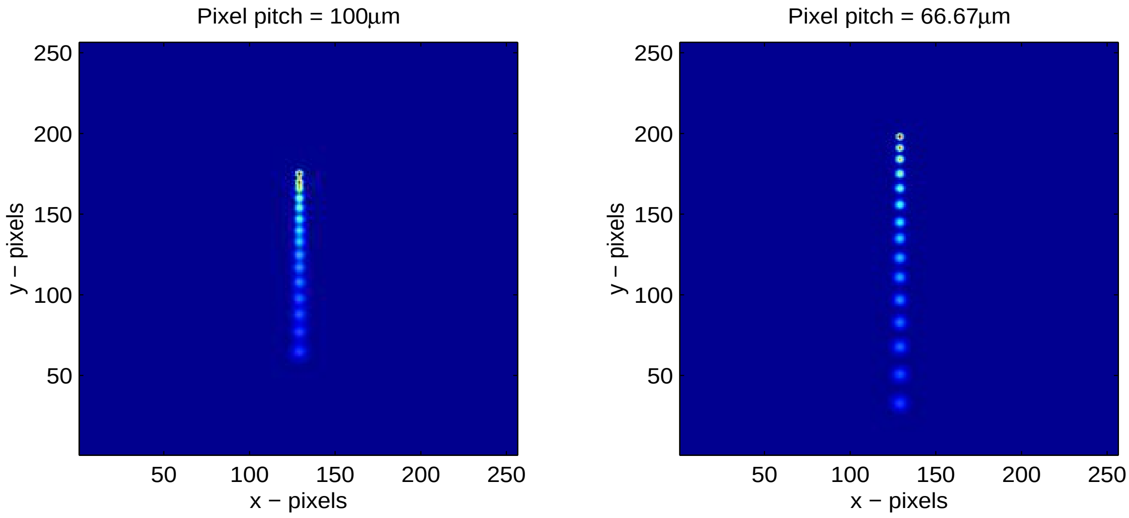


Figure 3.7: This figure shows the PSF at the detector for the vertical DVP rotation angle, $PSF_{\theta=0^o}$, for both a $66.67\mu m$ and $100\mu m$ detector pixel pitch.

It is evident how tailoring the extent of the detector PSF can be utilized for specific imaging scenario classes. The PSF resulting from the $100\mu m$ pixel pitch extends just under half the linear dimension of the detector. This allows an input image up to half the equivalent size of the detector to be collected when the image convolution with the PSF is conducted. Subsequent test cases in chapter IV, which are constructed to have a spatial extent of 100×100 pixels, will use this detector PSF.

Another application may require viewing a scene with a more limited spatial extent. The PSF resulting from the $66.67\mu m$ pixel pitch is spread further across the detector allowing a larger separation and thus better discrimination of consecutive spectral PSFs. However larger input images will be prone to projection off the edges of the detector. To simulate this imaging scenario several test cases in chapter IV are constructed with input hyperspectral data cubes which have a spatial extent of 20×20 pixels.

As shown by Goodman in chapter 2 of [10], the output of a LSI imaging system can be expressed as the 2D convolution of the input with the system transfer function. In the notation used in this thesis this is represented as

$$I(\theta_{DVP}, x_d, y_d) = PSF_\theta(\lambda, \theta_{DVP}, x_d, y_d) \otimes O(\lambda, u, v) \tag{3.12}$$

where $I(\theta_{DVP}, x_d, y_d)$ is a particular image at the detector resulting from the input object hyperspectral cube $O(\lambda, u, v)$.

3.4 Scene Object Image Reconstruction

Once the system obtains a particular image at the detector, $I(\theta_{DVP}, x_d, y_d)$, the goal is to then reconstruct the hyperspectral data cube which the system sensed to produce that image. This requires a reconstruction algorithm which has been previously implemented using deterministic linear algebra methods by Gustke in [11] and Mooney *et al* in [2, 3, 15, 16]. In an attempt to make use of physical aspects of the hyperspectral data cube and improve reconstruction results, this thesis implements an Estimation Theory based reconstruction algorithm. Further information on Estimation Theory is found in [21] by Van Trees.

3.4.1 Derivation of Two-Dimensional Reconstruction Algorithm. The derivation of the 2D reconstruction update equation begins with equation 3.12 and then explicitly states the image function as being dependent on both λ and θ. The convolution sum expands out as follows:

$$\begin{aligned} I(\theta_{DVP}, x_d, y_d) &= PSF_\theta(\lambda, \theta_{DVP}, x_d, y_d) \otimes O(\lambda, u, v) \\ i(\lambda, \theta, x, y) &= \sum_{u=1}^{N}\sum_{v=1}^{N}\sum_{\lambda=1}^{15} O(\lambda, u, v) PSF_\theta(\lambda, \theta, x-u, y-v) \end{aligned} \tag{3.13}$$

where the number of pixels on a detector side is $N = 256$, and λ is now treated as an index into a wavelength matrix and not the actual wavelength value. The notation of $i(\lambda, \theta, x, y)$ and $PSF_\theta(\lambda, \theta, x, y)$ also now explicitly show the dependence of the detector image and PSF on wavelength and the rotation angle of the DVP. The change to lower case also indicates that $i(\lambda, \theta, x, y)$ is variable; specifically a deterministic function of (λ, θ, x, y). The subscripts on (x_d, y_d) and θ_{DVP} are dropped to condense the notation.

It is then assumed that the photons arriving at the detector, or the data at the detector, $d(\lambda,\theta,x,y)$, is composed of a deterministic image resulting from equation 3.13 plus random background noise, $n(\lambda,\theta,x,y)$. This can be expressed as

$$d(\lambda,\theta,x,y) = i(\lambda,\theta,x,y) + n(\lambda,\theta,x,y) \tag{3.14}$$

The addition of a deterministic variable and a random variable results in a sum which is itself a random variable. Hence $d(\lambda,\theta,x,y)$ is regarded as a random variable. Note that in this notation a lower case letter indicates a random or deterministic variable while the corresponding upper case letter denotes a realization of that variable. For example, D is a particular realization of the random variable d.

The physics of photon noise can be modelled as a Poisson random variable as indicated by Dereniak and Boreman in chapter 5 of [7]. As shown by Papoulis and Pillai in chapter 4 of [17] the basic form of the Poisson probability mass function (PMF) is

$$P[w = k] = \frac{\beta^k \ e^{-\beta}}{k!} \tag{3.15}$$

where w is a discrete random variable, $k = \{0,\ 1,\ 2,\ ...,\ \infty\}$, and β is equal to both the mean and variance of w. As such the arrival of photons at the detector can be modelled as a Poisson random variable with a mean equal to the deterministic image, $i(\lambda,\theta,x,y)$, expressed as

$$\beta = E[d(\lambda,\theta,x,y)] = i(\lambda,\theta,x,y) \tag{3.16}$$

Note that for this application the realizations of the Poisson random variables are independent and identically distributed (iid).

The probability that the data, or number of photons, at one DVP rotation angle, θ_o, at one pixel in the detector, (x_o, y_o), can then be found by equating equation 3.16 with the Poisson PMF at equation 3.15 as

$$P[d(\lambda, \theta, x, y) = D(\theta_o, x_o, y_o)] = \frac{i(\lambda, \theta_o, x_o, y_o)^{D(\theta_o, x_o, y_o)} e^{-i(\lambda, \theta_o, x_o, y_o)}}{D(\theta_o, x_o, y_o)!} \tag{3.17}$$

where $D(\theta_o, x_o, y_o)$ is a realization of the random variable $d(\lambda, \theta, x, y)$ at the pixel (x_o, y_o) at DVP rotation angle θ_o.

Equation 3.17 will hold for every pixel in the detector at each DVP rotation angle. As photon arrival is iid, the expression for the entire probability of photons in the data cube at the detector can be expressed as

$$P[d = D \; \forall \; x, y, \theta] = \prod_{x=1}^{256} \prod_{y=1}^{256} \prod_{\theta=1}^{15} \frac{i(\lambda, \theta, x, y)^{D(\theta, x, y)} e^{-i(\lambda, \theta, x, y)}}{D(\theta, x, y)!} \tag{3.18}$$

which is the product of the probabilities for the individual pixels at each DVP rotation angle.

In order to maximize the probability of photons at the detector the maxima of equation 3.18 needs to be determined. To ease the formulation of the derivative the natural logarithm of both sides of equation 3.18 is taken. As the natural logarithm is a monotonically increasing function, maximizing $ln\{P[d = D \; \forall \; x, y, \theta]\}$ will also maximize $P[d = D \; \forall \; x, y, \theta]$. Accordingly, the natural logarithm of equation 3.18 is

$$\begin{aligned} ln\{P[d = D]\} &= \sum_{x=1}^{256} \sum_{y=1}^{256} \sum_{\theta=1}^{15} ln\left\{i(\lambda, \theta, x, y)^{D(\theta, x, y)}\right\} + ln\left\{e^{-i(\lambda, \theta, x, y)}\right\} - ln\{D(\theta, x, y)!\} \\ &= \sum_{x=1}^{256} \sum_{y=1}^{256} \sum_{\theta=1}^{15} D(\theta, x, y) ln\{i(\lambda, \theta, x, y)\} - i(\lambda, \theta, x, y) - ln\{D(\theta, x, y)!\} \end{aligned} \tag{3.19}$$

where the logarithm relationships: $log_a x^r = rlog_a x$; $log_a xy = log_a x + log_a y$; and $log_a \frac{x}{y} = log_a x - log_a y$ from [20] are applicable.

The photon probability at the detector, given a point in an input hyperspectral data cube $O(\lambda_o, u_o, v_o)$, is now maximized by taking the derivative of equation 3.19 as follows:

$$\frac{d\ ln\{P[d = D]\}}{d\ O(\lambda_o, u_o, v_o)} = \sum_{x=1}^{256}\sum_{y=1}^{256}\sum_{\theta=1}^{15} \frac{D(\theta, x, y)}{i(\lambda, \theta, x, y)} \frac{d\ i(\lambda, \theta, x, y)}{d\ O(\lambda_o, u_o, v_o)} - \frac{d\ i(\lambda, \theta, x, y)}{d\ O(\lambda_o, u_o, v_o)} \tag{3.20}$$

Note that the derivative associated with the $-ln\{D(\theta, x, y)!\}$ term goes to zero as D, being a realization of the random variable d, is a constant.

Taking the derivative of equation 3.13 with respect to $O(\lambda_o, u_o, v_o)$ results in

$$\begin{aligned}\frac{d\ i(\lambda, \theta, x, y)}{d\ O(\lambda_o, u_o, v_o)} &= \frac{d}{d\ O(\lambda_o, u_o, v_o)} \sum_{u=1}^{N}\sum_{v=1}^{N}\sum_{\theta=1}^{15} O(\lambda, u, v) PSF_\theta(\lambda, \theta, x - u, y - v) \\ &= \frac{d}{d\ O(\lambda_o, u_o, v_o)} O(\lambda_o, u_o, v_o) PSF_\theta(\lambda_o, \theta, x - u_o, y - v_o) \\ &= PSF_\theta(\lambda_o, \theta, x - u_o, y - v_o)\end{aligned} \tag{3.21}$$

as the derivative is zero where $(\lambda, u, v) \neq (\lambda_o, u_o, v_o)$.

Substitution of equation 3.21 into equation 3.20 yields

$$\frac{d\ ln\{P[d = D]\}}{d\ O(\lambda_o, u_o, v_o)} = \sum_{x=1}^{256}\sum_{y=1}^{256}\sum_{\theta=1}^{15} \frac{D(\theta, x, y)}{i(\lambda, \theta, x, y)} \times PSF_\theta(\lambda_o, \theta, x - u_o, y - v_o) - \frac{d\ i(\lambda, \theta, x, y)}{d\ O(\lambda_o, u_o, v_o)} \tag{3.22}$$

Also observe from equation 3.21 that

$$\begin{aligned}\sum_{x=1}^{256}\sum_{y=1}^{256}\sum_{\theta=1}^{15}\frac{d\ i(\lambda,\theta,x,y)}{d\ O(\lambda_o,u_o,v_o)} &= \sum_{x=1}^{256}\sum_{y=1}^{256}\sum_{\theta=1}^{15} PSF_\theta(\lambda_o,\theta,x-u_o,y-v_o) \\ &= 15\equiv\Lambda\end{aligned}\tag{3.23}$$

where Λ= 15, which is equivalent to the number of spectral bins.

Upon substituting equation 3.23 into equation 3.22 the result becomes

$$\frac{d\ ln\{P[d=D]\}}{d\ O(\lambda_o,u_o,v_o)} = \sum_{x=1}^{256}\sum_{y=1}^{256}\sum_{\theta=1}^{15}\frac{D(\theta,x,y)}{i(\lambda,\theta,x,y)}\times PSF_\theta(\lambda_o,\theta,x-u_o,y-v_o)-\Lambda\tag{3.24}$$

The derivative is set to zero and Λ is added to both sides resulting in

$$\Lambda = \sum_{x=1}^{256}\sum_{y=1}^{256}\sum_{\theta=1}^{15}\frac{D(\theta,x,y)}{i(\lambda,\theta,x,y)}\times PSF_\theta(\lambda_o,\theta,x-u_o,y-v_o)\tag{3.25}$$

Both sides of equation 3.25 are multiplied by an estimate of the complete hyperspectral data cube $\widehat{O}(\lambda,u,v)$ and divided by Λ to obtain

$$\widehat{O}(\lambda,u,v) \quad = \quad \frac{\widehat{O}(\lambda,u,v)}{\Lambda}\sum_{x=1}^{256}\sum_{y=1}^{256}\sum_{\theta=1}^{15}\frac{D(\theta,x,y)}{i(\lambda,\theta,x,y)}\times PSF_\theta(\lambda,\theta,x-u,y-v)\tag{3.26}$$

It is now observed that the 3D sum over (x,y,θ) has the form of an autocorrelation as shown by Goodman in chapter 2 of [10]. Thus the Autocorrelation Theorem of Fourier Transform theory can be applied in the implementation of equation 3.26 as an iterative update equation. The

Autocorrelation Theorem states that

$$\begin{aligned}
\mathcal{F}\left\{\iint\limits_{-\infty}^{\infty} g(\xi,\eta)h^{*}(\xi - x,\eta - y)\ d\xi d\eta\right\} &= G(f_x,f_y)H^{*}(f_x,f_y) \\
&\text{and} \\
\mathcal{F}\{g(x,y)h^{*}(x,y)\} &= \iint\limits_{-\infty}^{\infty} G(\xi,\eta)H^{*}(\xi - f_x,\eta - f_y)\ d\xi d\eta
\end{aligned} \tag{3.27}$$

where $\mathcal{F}$ indicates a Fourier Transform between the transform pairs of $g(\xi,\eta)$ and $G(f_x,f_y)$, and also $h(x,y)$ and $H(f_x,f_y)$.

As previously discussed the scene object hyperspectral cube, $O(\lambda,u,v)$, occurs over two spatial dimensions and one wavelength dimension. However the data at the detector, $d(\lambda,\theta,x,y)$, is a function of four dimensions, with the DVP rotation angle being the additional dimension. Thus it becomes necessary for the iterative implementation of equation 3.26 to be conducted in two stages as shown in Figure 3.8. The algorithm begins with an initial estimate of O which is labelled $\widehat{O}$ and arbitrarily set to all ones. The algorithm then makes use of the Autocorrelation Theorem at equation 3.27 by summing the product of each $\mathcal{F}\{\widehat{O}\}$ wavelength slice with the corresponding

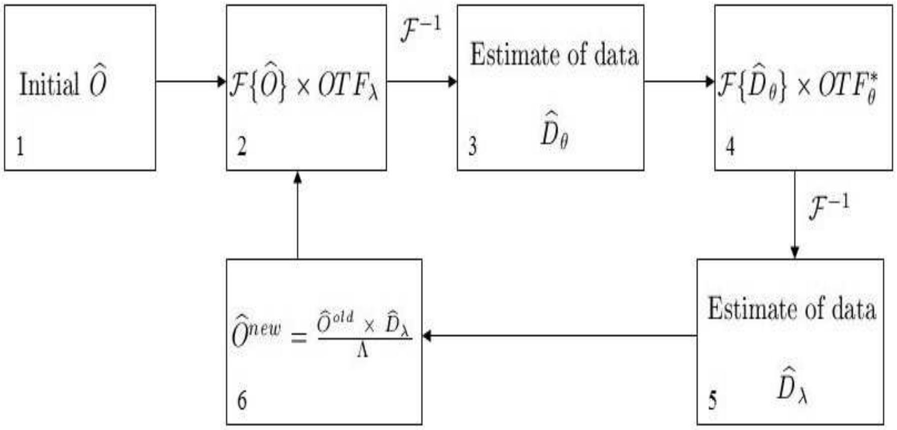

Initial $\widehat{O}$
1
$\mathcal{F}\{\widehat{O}\} \times OTF_{\lambda}$
2
$\mathcal{F}^{-1}$
Estimate of data
$\widehat{D}_{\theta}$
3
$\mathcal{F}\{\widehat{D}_{\theta}\} \times OTF^{*}_{\theta}$
4
$\mathcal{F}^{-1}$
Estimate of data
$\widehat{D}_{\lambda}$
5
$\widehat{O}^{new} = \frac{\widehat{O}^{old} \times \widehat{D}_{\lambda}}{\Lambda}$
6

Figure 3.8: This figure shows the processing steps required in implementing equation 3.26 as an iterative estimation algorithm for reconstructing the scene hyperspectral data cube.

OTF_λ slice. Recall that the OTF is the Fourier Transform of the intensity PSF. Thus OTF_λ is the Fourier Transform of PSF_θ summed over each rotation angle of the DVP, ie,

$$OTF_\lambda(\lambda, x, y) = \mathcal{F}\left\{\sum_{\theta=1}^{15} PSF_\theta(\lambda, \theta, x, y)\right\} \tag{3.28}$$

where θ is indexing the DVP rotation angle positions. The Inverse Fourier Transform ($\mathcal{F}^{-1}$) of this product provides an estimate of the data, $\widehat{D}_\theta$, at each DVP rotation angle. An estimate of the data in each wavelength bin, $\widehat{D}_\lambda$, is then found by summing the product of Fourier Transform of each DVP rotation angle slice of $\widehat{D}_\theta$ with the complex conjugate of each OTF_θ slice where

$$OTF_\theta(\theta, x, y) = \mathcal{F}\left\{\sum_{\lambda=1}^{15} PSF_\theta(\lambda, \theta, x, y)\right\} \tag{3.29}$$

which is equivalent to the Fourier Transform of the detector PSF shown in Figure 3.7. The dimensions of the estimate of the hyperspectral cube object, $\widehat{O}$, and the estimate of the data at the detector, $\widehat{D}_\lambda$, now correspond and the algorithm can be updated with a new $\widehat{O}$ estimate where

$$\widehat{O}^{new} = \frac{\widehat{O}^{old} \ \times \ \widehat{D}_\lambda}{\Lambda} \tag{3.30}$$

Thus equation 3.26 is implemented in an iterative estimation algorithm where the scene object hyperspectral data cube as an update equation of

$$\widehat{O}^{new}(\lambda, u, v) \quad = \quad \frac{\widehat{O}^{old}(\lambda, u, v)}{\Lambda} \sum_{x=1}^{256}\sum_{y=1}^{256}\sum_{\theta=1}^{15} \frac{\widehat{D}(\theta, x, y)}{i(\lambda, \theta, x, y)} \times PSF_\theta(\lambda, \theta, x-u, y-v) \tag{3.31}$$

3.4.2 Derivation of Vector Reconstruction Algorithm. In some hyperspectral imaging applications it may be sufficient to have only one-dimensional (1D) spatial information in the reconstructed hyperspectral scene. Using a 1D vector of the detector image for scene reconstruction offers significant processing time improvements in both collecting the detector data and within the reconstruction algorithm itself. When paired with the CTIS which inherently allows for fast data collection, this approach could potentially be applied to imaging a temporally evolving scene and also some real-time applications.

The choice of how to turn the 2D detector image into a vector takes advantage of the in-built functionality available in some infrared (IR) detector arrays. The vector is formed by summing the image in the columns (or rows) of the detector image. The follow-on work in this thesis will be applicable to the sum of each column in the detector which forms a horizontal, or x, vector sum of the detector image. However it is also observed that with either an instrument modification which includes a beam-splitter and additional DOS offset at right-angles, or a phasing between vector collection, both the horizontal and vertical vectors could be collected. This would allow collection of pseudo-2D detector data permitting the reconstruction of more spatial scene information. The x vector sum resulting from the 2D $66.67\mu m$ pixel pitch PSF in Figure 3.7 is shown in Figure 3.9.

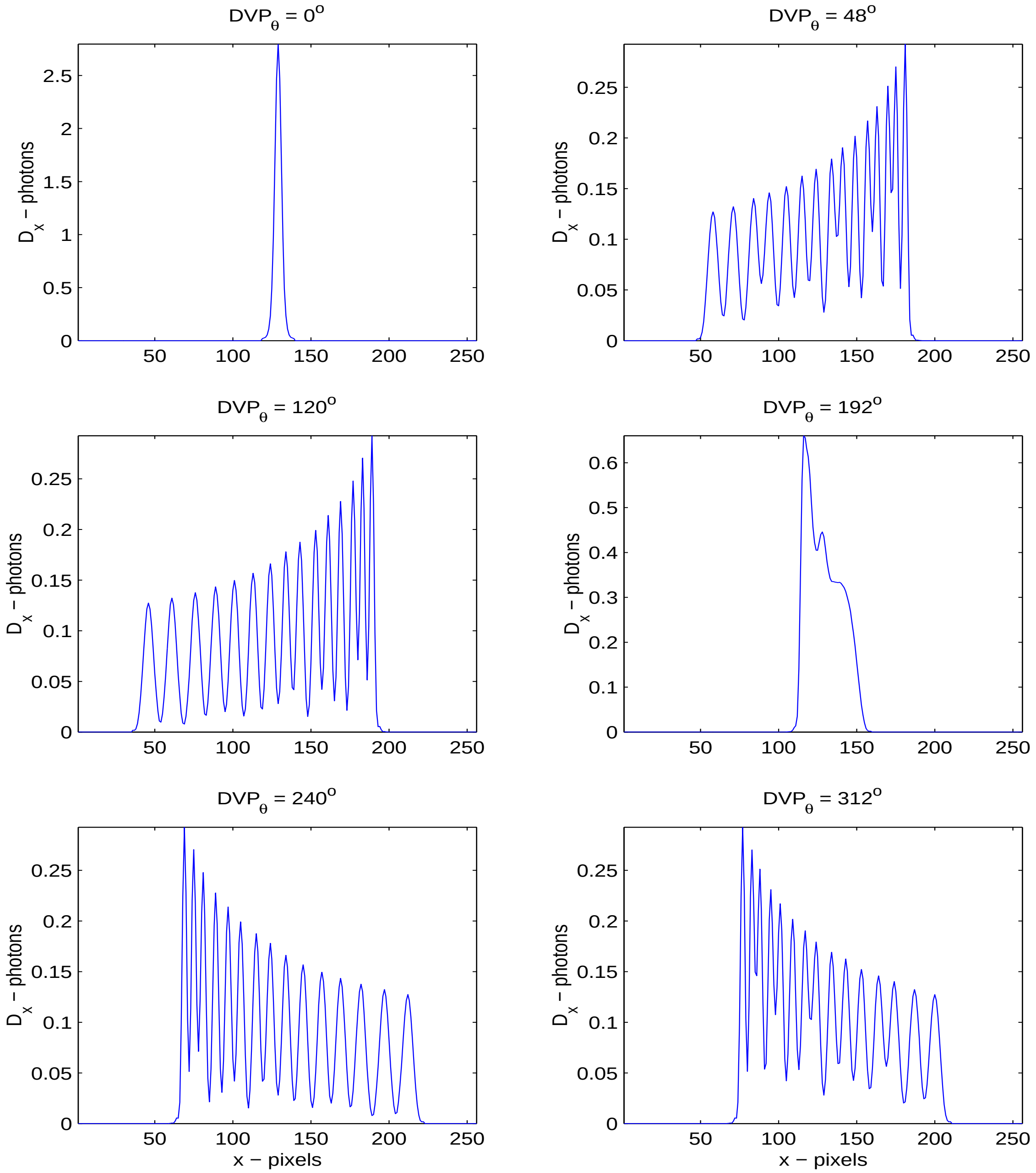


Figure 3.9: This figure shows the x vector sum at six DVP rotation angles of the detector PSF for the $66.67\mu m$ detector pixel pitch.

The derivation of the vector reconstruction algorithm begins with summing along the vertical, or y, direction of equation 3.14 which results in

$$\begin{aligned} d(\lambda,\theta,x,y) &= i(\lambda,\theta,x,y)+n(\lambda,\theta,x,y) \\ \sum_{y=1}^{256} d(\lambda,\theta,x,y) &= \sum_{y=1}^{256}\{i(\lambda,\theta,x,y)+n(\lambda,\theta,x,y)\} \\ d_x(\lambda,\theta,x) &= i_x(\lambda,\theta,x)+n_x(\lambda,\theta,x) \end{aligned} \tag{3.32}$$

where $d_x(\lambda,\theta,x)$, $i_x(\lambda,\theta,x)$ and $n_x(\lambda,\theta,x)$ are vector sums of the corresponding 2D detector parameters.

The mean of the vector detector data is again expressed as

$$\beta = E[d_x(\lambda,\theta,x)] = i_x(\lambda,\theta,x) \tag{3.33}$$

which can be substituted into the Poisson PMF at equation 3.15 resulting in

$$P[d_x(\lambda,\theta,x) = D_x(\theta_o,x_o)] = \frac{i_x(\lambda,\theta_o,x_o)^{D_x(\theta_o,x_o)}e^{-i_x(\lambda,\theta_o,x_o)}}{D_x(\theta_o,x_o)!} \tag{3.34}$$

where $D_x(\theta_o,x_o)$ is a realization of the random variable $d_x(\lambda,\theta,x)$ at the pixel x_o and DVP rotation angle θ_o. The probability of photon arrival at each pixel is iid. Hence when considering all the pixels in the vector at each DVP rotation angle the total probability becomes

$$P[d_x = D_x \;\forall\; x,\theta] = \prod_{x=1}^{256}\prod_{\theta=1}^{15}\frac{i_x(\lambda,\theta,x)^{D_x(\theta,x)}e^{-i_x(\lambda,\theta,x)}}{D_x(\theta,x)!} \tag{3.35}$$

which is the product of the probabilities for the individual pixels at each DVP rotation angle.

The maxima of equation 3.35 needs to be determined to maximize the probability of photons at the detector. This is again performed by firstly taking the natural logarithm of both sides of equation 3.35 as

$$ln\{P[d_x = D_x]\} \quad = \quad \sum_{x=1}^{256}\sum_{\theta=1}^{15} D_x(\theta,x) ln\{i_x(\lambda,\theta,x)\} - i_x(\lambda,\theta,x) - ln\{D_x(\theta,x)!\} \tag{3.36}$$

The photon probability given a point in an input hyperspectral data matrix $O(\lambda_o, u_o)$ is now maximized by taking the derivative of equation 3.36 as follows:

$$\frac{d\ ln\{P[d_x = D_x]\}}{d\ O(\lambda_o,u_o)} = \sum_{x=1}^{256}\sum_{\theta=1}^{15} \frac{D_x(\theta,x)}{i_x(\lambda,\theta,x)} \frac{d\ i_x(\lambda,\theta,x)}{d\ O(\lambda_o,u_o)} - \frac{d\ i_x(\lambda,\theta,x)}{d\ O(\lambda_o,u_o)} \tag{3.37}$$

where $O(\lambda_o, u_o)$ is constructed from $O(\lambda_o, u_o, v_o)$ by summing along the v_o direction.

In order to include a similar substitution as used in the 2D algorithm the detector PSF must again be examined. Proceeding from equation 3.13 of the 2D case the dimension reduction on i, O and PSF_θ begins with summing both sides over y to produce the x vector sum image $i_x(\lambda,\theta,x)$ as

$$\begin{aligned}
i(\lambda,\theta,x,y) &= \sum_{u=1}^{256}\sum_{v=1}^{256}\sum_{\lambda=1}^{15} O(\lambda,u,v) PSF_\theta(\lambda,\theta,x-u,y-v) \\
\sum_{y=1}^{256} i(\lambda,\theta,x,y) &= \sum_{y=1}^{256}\left\{\sum_{u=1}^{256}\sum_{v=1}^{256}\sum_{\lambda=1}^{15} O(\lambda,u,v) PSF_\theta(\lambda,\theta,x-u,y-v)\right\} \\
i_x(\lambda,\theta,x) &= \sum_{y=1}^{256}\left\{\sum_{u=1}^{256}\sum_{v=1}^{256}\sum_{\lambda=1}^{15} O(\lambda,u,v) PSF_\theta(\lambda,\theta,x-u,y-v)\right\} \qquad (3.38)
\end{aligned}$$

The Fourier Transform of both sides of equation 3.38 is then performed turning the convolution within the sum into a product of Fourier Transforms as

$$\mathcal{F}\{i_x(\lambda,\theta,x)\} = \sum_{y=1}^{256} \mathcal{F}\{O(\lambda,u,v)\} \times OTF_\theta(\lambda,\theta,u,v) \tag{3.39}$$

where the Fourier Transform of the PSF is the OTF and the Convolution Theorem is utilized.

The Inverse Fourier Transform of equation 3.39 is then written explicitly as

$$\begin{aligned} i_x(\lambda,\theta,x) &= \sum_{u=1}^{256}\sum_{v=1}^{256}\left[\sum_{y=1}^{256}\mathcal{F}\{O(\lambda,u,v)\}\times OTF_\theta(\lambda,\theta,u,v)\right]e^{\frac{j2\pi(xu+yv)}{N}} \\ &= \sum_{u=1}^{256}\sum_{v=1}^{256}\left[\mathcal{F}\{O(\lambda,u,v)\}\times OTF_\theta(\lambda,\theta,u,v)\right]e^{\frac{j2\pi(xu)}{N}}\sum_{y=1}^{256}1e^{\frac{j2\pi(yv)}{N}} \end{aligned} \tag{3.40}$$

where the order of the sums is modified and the exponential product is separated. Note that the sum of y is now separable and is the Inverse Fourier Transform of a constant which results in

$$i_x(\lambda,\theta,x) = \sum_{u=1}^{256}\sum_{v=1}^{256}\left[\mathcal{F}\{O(\lambda,u,v)\}\times OTF_\theta(\lambda,\theta,u,v)\right]e^{\frac{j2\pi(xu)}{N}}\ \delta[v] \tag{3.41}$$

where $\delta[v]$ is a modified discrete Kronecker delta function with

$$\begin{aligned} \delta[v] &= 1 \text{ for } v=1 \\ &= 0 \text{ for } v\neq 1 \end{aligned} \tag{3.42}$$

The delta function serves to eliminate the sum over v as

$$\begin{aligned} i_x(\lambda,\theta,x) &= \left[\sum_{v=1}^{256}\delta[v]\right]\sum_{u=1}^{256}\left[\mathcal{F}\{O(\lambda,u,v)\}\times OTF_\theta(\lambda,\theta,u,v)\right]e^{\frac{j2\pi(xu)}{N}} \\ &= \sum_{u=1}^{256}\left[\mathcal{F}\{O(\lambda,u,1)\}\times OTF_\theta(\lambda,\theta,u,1)\right]e^{\frac{j2\pi(xu)}{N}} \\ &= \sum_{u=1}^{256}\left[\mathcal{F}\{O(\lambda,u,1)\}\times OTF_{\theta v}(\lambda,\theta,u)\right]e^{\frac{j2\pi(xu)}{N}} \end{aligned} \tag{3.43}$$

where $OTF_{\theta v}(\lambda, \theta, u) = OTF_{\theta}(\lambda, \theta, u, 1)$. Note that the 3D OTF reduces to the baseband slice at $v = 1$ of the four dimensional (4D) OTF. This is alternately viewed as the main horizontal axis of spatial image.

The Inverse Fourier Transform of the right-hand-side (RHS) of equation 3.43 is then taken to produce the final expression for $i_x(\lambda, \theta, x)$ as

$$\begin{aligned} i_x(\lambda, \theta, x) &= \sum_{u=1}^{256}\sum_{\lambda=1}^{15} O(\lambda, u, 1) PSF_{\theta}(\lambda, \theta, x-u, 1) \\ &= \sum_{u=1}^{256}\sum_{\lambda=1}^{15} O_x(\lambda, u) PSF_{\theta x}(\lambda, \theta, x-u) \end{aligned} \tag{3.44}$$

where $O_x(\lambda, u) = O(\lambda, u, 1)$ and $PSF_{\theta x}(\lambda, \theta, x)$ is a vector representation of the 2D detector PSF at the baseband axis. The implementation of the vector reconstruction algorithm uses the horizontal cross-section of the 2D OTF at the center of the detector as $OTF_{\theta x}(\lambda, \theta, u)$. The Inverse Fourier Transform of $OTF_{\theta v}(\lambda, \theta, u)$ then provides $PSF_{\theta x}(\lambda, \theta, x)$.

Similar to equation 3.24 in the 2D case, equation 3.37 becomes

$$\frac{d\; ln\{P[d_x = D_x]\}}{d\; O(\lambda_o, u_o)} = \sum_{x=1}^{256}\sum_{\theta=1}^{15} \frac{D_x(\theta, x)}{i_x(\lambda, \theta, x)} \times PSF_{\theta x}(\lambda_o, \theta, x-u_o) - \Lambda_x \tag{3.45}$$

for the vector case when maximizing the photon probability for a given an input $O(\lambda_o, u_o)$ where $\Lambda_x = 15$.

In an identical approach to the 2D reconstruction algorithm, equation 3.45 can be implemented as an iterative vector reconstruction algorithm using

$$\widehat{O}^{new}(\lambda, u) = \frac{\widehat{O}^{old}(\lambda, u)}{\Lambda_x} \sum_{x=1}^{256}\sum_{\theta=1}^{15} \frac{\widehat{D}_x(\theta, x)}{i_x(\lambda, \theta, x)} \times PSF_{\theta x}(\lambda, \theta, x-u) \tag{3.46}$$

3.4.3 Methods for Including Atmospheric Transmission Coefficients. Light at IR wavelengths propagating through the atmosphere will experience wavelength dependent attenuation. This attenuation is characterized by the atmospheric transmission coefficient, denoted $t_{atm}(\lambda)$, with values as shown in Figure 3.10.

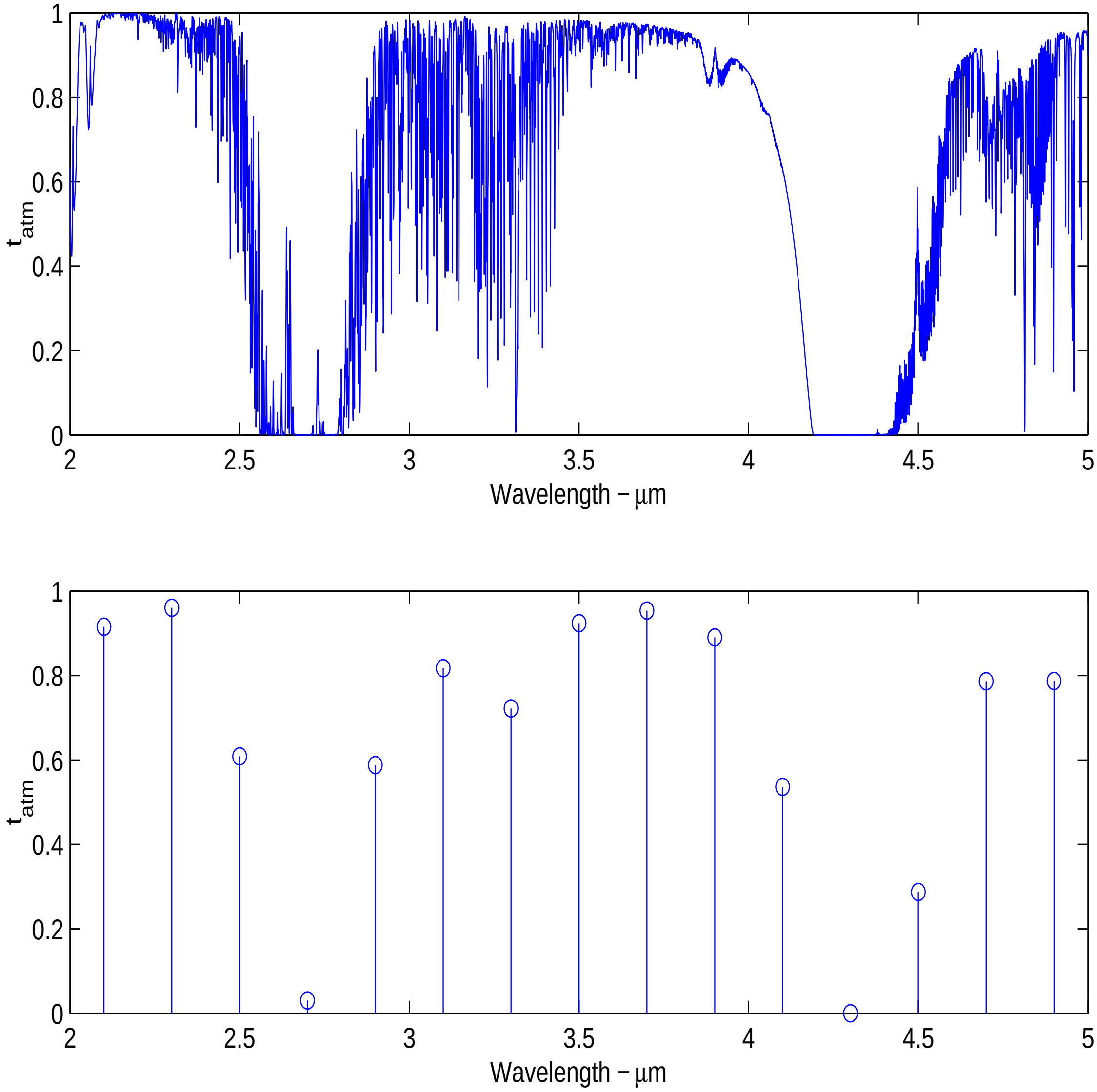


Figure 3.10: This figure shows the wavelength dependent atmospheric transmission coefficient, $t_{atm}(\lambda)$, with the pseudo-continuous data in the upper subplot taken from [14]. This data is then sampled by averaging over each wavelength bin as shown in the lower subplot.

The pseudo-continuous data for $t_{atm}(\lambda)$ is available at Gemini Observatory website at [14], where the data is applicable for an observer at an altitude of 20 kilometers looking straight down at the Earth. This data is then sampled by averaging over each wavelength bin for subsequent inclusion into the reconstruction algorithms with the discrete $t_{atm}(\lambda)$ for 15 wavelength bins shown in Table 3.3.

Table 3.3: Discrete Atmospheric Transmission Coefficients ($t_{atm}(\lambda)$)

Bin	**Bin Center**	$t_{atm}(\lambda)$	**Bin**	**Bin Center**	$t_{atm}(\lambda)$	**Bin**	**Bin Center**	$t_{atm}(\lambda)$
λ_i	(μm)		λ_i	(μm)		λ_i	(μm)	
1	2.1	0.9158	6	3.1	0.8177	11	4.1	0.5365
2	2.3	0.9603	7	3.3	0.7221	12	4.3	0.0003
3	2.5	0.6089	8	3.5	0.9243	13	4.5	0.2875
4	2.7	0.0308	9	3.7	0.9535	14	4.7	0.7868
5	2.9	0.5881	10	3.9	0.8903	15	4.9	0.7874

There are two approaches for including $t_{atm}(\lambda)$ in the reconstruction as follows:

3.4.3.1 Direct Inversion of $t_{atm}(\lambda)$.

This method first uses the algorithm in either section 3.4.1 or 3.4.2 to reconstruct the input scene hyperspectral data cube. It is then assumed that the number of photons in each wavelength bin has been attenuated by the corresponding $t_{atm}(\lambda)$. Hence the reconstruction algorithm has estimated an atmospherically attenuated input data cube. Thus in order to find the actual input data cube each wavelength bin of the reconstructed data cube is divided by $t_{atm}(\lambda)$ as

$$\widehat{O}(\lambda, u, v) = \frac{\widehat{O}_{atm}(\lambda, u, v)}{t_{atm}(\lambda)} \tag{3.47}$$

where $\widehat{O}_{atm}(\lambda, u, v)$ is the output from the reconstruction algorithm and $t_{atm}(\lambda)$ is the respective atmospheric coefficient for each wavelength bin. Though simple in its implementation, this method is prone to error which increases as $t_{atm}(\lambda)$ decreases, ie, error increases in the bins with more atmospheric attenuation.

3.4.3.2 Inclusion of $t_{atm}(\lambda)$ within the reconstruction algorithm. This method includes the atmospheric attenuation information within the reconstruction algorithm. This requires rederivation of the algorithm to include $t_{atm}(\lambda)$. Though equally applicable to both the 2D and vector reconstruction algorithm, only the derivation for the attenuated x vector reconstruction algorithm is explicitly shown.

This derivation proceeds by modifying data at the detector at equation 3.32 as

$$d_x(\lambda, \theta, x) \quad = \quad i_x(\lambda, \theta, x) \times t_{atm}(\lambda) + n_x(\lambda, \theta, x) \tag{3.48}$$

which recognizes that the deterministic image at the detector, $i_x(\lambda, \theta, x)$, has been subject to a wavelength dependent atmospheric attenuation $t_{atm}(\lambda)$.

The mean of the vector detector data is now expressed as

$$\beta = E[d_x(\lambda, \theta, x)] = i_x(\lambda, \theta, x) \times t_{atm}(\lambda) \tag{3.49}$$

which can be substituted into the Poisson PMF at equation 3.15 resulting in

$$P[d_x(\lambda, \theta, x) = D_x(\theta_o, x_o)] = \frac{[i_x(\lambda, \theta_o, x_o)t_{atm}(\lambda)]^{D_x(\theta_o, x_o)} e^{[-i_x(\lambda, \theta_o, x_o)t_{atm}(\lambda)]}}{D_x(\theta_o, x_o)!} \tag{3.50}$$

When extended to all the pixels in the vector and given that the photon probability of the pixels are idd, the total probability becomes

$$P[d_x = D_x \ \forall \ x, \theta] = \prod_{x=1}^{256} \prod_{\theta=1}^{15} \frac{[i_x(\lambda, \theta, x)t_{atm}(\lambda)]^{D_x(\theta, x)} e^{[-i_x(\lambda, \theta, x)t_{atm}(\lambda)]}}{D_x(\theta, x)!} \tag{3.51}$$

which is the product of the probabilities for the individual pixels and wavelength bins.

The maxima of equation 3.51 must be determined to maximize the probability of photons at the detector. This is again performed by firstly taking the natural logarithm of both sides of equation 3.51 as

$$\begin{aligned} ln\{P[d_x = D_x]\} &= \sum_{x=1}^{256}\sum_{\theta=1}^{15} D_x(\theta,x) ln\{i_x(\lambda,\theta,x)t_{atm}(\lambda)\} - i_x(\lambda,\theta,x)t_{atm}(\lambda) - ln\{D_x(\theta,x)!\} \\ &= \sum_{x=1}^{256}\sum_{\theta=1}^{15} D_x(\theta,x) ln\{i_x(\lambda,\theta,x)\} + D_x(\theta,x) ln\{t_{atm}(\lambda)\} \cdots \\ & \quad -i_x(\lambda,\theta,x)t_{atm}(\lambda) - ln\{D_x(\theta,x)!\} \end{aligned} \tag{3.52}$$

The photon probability given a point in an input hyperspectral data matrix $O(\lambda_o, u_o)$ is now maximized by taking the derivative of equation 3.52 as follows:

$$\begin{aligned} \frac{d\ ln\{P[d_x = D_x]\}}{d\ O(\lambda_o,u_o)} &= \sum_{x=1}^{256}\sum_{\theta=1}^{15} \frac{D_x(\theta,x)}{i_x(\lambda,\theta,x)}\frac{d\ i_x(\lambda,\theta,x)}{d\ O(\lambda_o,u_o)} + \frac{D_x(\theta,x)}{t_{atm}(\lambda)}\frac{d\ t_{atm}(\lambda)}{d\ O(\lambda_o,u_o)} - \frac{d\ i_x(\lambda,\theta,x)t_{atm}(\lambda)}{d\ O(\lambda_o,u_o)} \\ &= \sum_{x=1}^{256}\sum_{\theta=1}^{15} \frac{D_x(\theta,x)}{i_x(\lambda,\theta,x)}\frac{d\ i_x(\lambda,\theta,x)}{d\ O(\lambda_o,u_o)} - t_{atm}(\lambda)\frac{d\ i_x(\lambda,\theta,x)}{d\ O(\lambda_o,u_o)} \end{aligned} \tag{3.53}$$

as $\frac{d\ t_{atm}(\lambda)}{d\ O(\lambda_o,u_o)} = 0\ \forall\ \lambda$ as $t_{atm}(\lambda)$ is a constant for each wavelength bin.

The substitution of

$$\frac{d\ i_x(\lambda,\theta,x)}{d\ O(\lambda_o,u_o)} = PSF_{\theta x}(\lambda,\theta,x-u_o) \tag{3.54}$$

is now applied to equation 3.53 resulting in

$$\begin{aligned} \frac{d\ ln\{P[d_x = D_x]\}}{d\ O(\lambda_o,u_o)} &= \sum_{x=1}^{256}\sum_{\theta=1}^{15} \frac{D_x(\theta,x)}{i_x(\lambda,\theta,x)} \times PSF_{\theta x}(\lambda,\theta,x-u_o) - t_{atm}(\lambda)PSF_{\theta x}(\lambda,\theta,x-u_o) \\ &= \sum_{x=1}^{256}\sum_{\theta=1}^{15} \frac{D_x(\theta,x)}{i_x(\lambda,\theta,x)} \times PSF_{\theta x}(\lambda,\theta,x-u_o) - \Upsilon \end{aligned} \tag{3.55}$$

where $\Upsilon = \sum_{\lambda=1}^{15} t_{atm}(\lambda)$, ie, the sum of each atmospheric transmission coefficient. Note that the maximum possible value of each $t_{atm}(\lambda)$ is 1 so the feasible values of $0 < \Upsilon \leq \Lambda$ apply. As an example, $\Upsilon \approx 9.8$ for the 15 wavelength bins shown in Figure 3.10.

The atmospherically attenuated vector reconstruction algorithm can be implemented by using equation 3.55 to modify the update equation of the iterative vector reconstruction algorithm as

$$\widehat{O}^{new}(\lambda, u) = \frac{\widehat{O}^{old}(\lambda, u)}{\Upsilon} \sum_{x=1}^{256} \sum_{\theta=1}^{15} \frac{\widehat{D}_x(\theta, x)}{i_x(\lambda, \theta, x)} \times PSF_{\theta x}(\lambda, \theta, x - u) \tag{3.56}$$

with modifications to the algorithm flow diagram as shown in Figure 3.11.

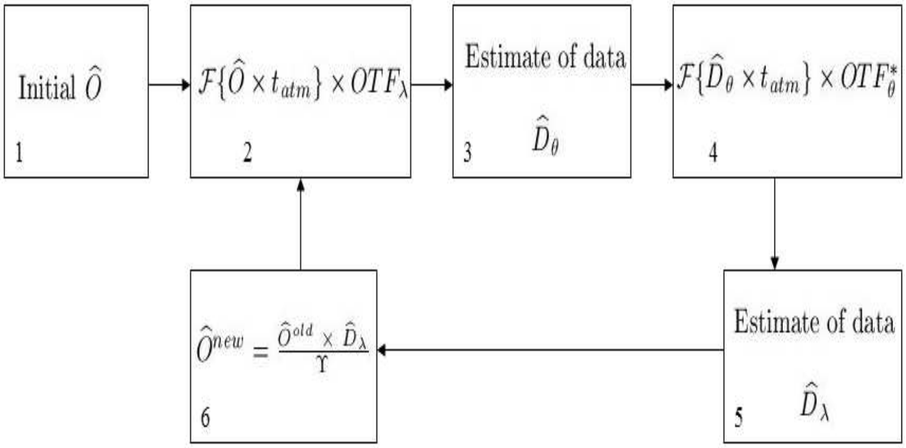

Initial $\widehat{O}$
1
$\mathcal{F}\{\widehat{O} \times t_{atm}\} \times OTF_\lambda$
2
Estimate of data
3
$\widehat{D}_\theta$
$\mathcal{F}\{\widehat{D}_\theta \times t_{atm}\} \times OTF_\theta^*$
4
Estimate of data
5
$\widehat{D}_\lambda$
$\widehat{O}^{new} = \frac{\widehat{O}^{old} \times \widehat{D}_\lambda}{\Upsilon}$
6

Figure 3.11: This figure shows the processing steps required in implementing equation 3.56 as an iterative estimation algorithm for reconstructing the atmospherically attenuated scene hyperspectral data cube.

IV. Test Case Scenarios

This chapter establishes the test cases used to verify the performance of the reconstruction algorithms in section 3.4. For each test case an input hyperspectral data cube for the imaging scenario is generated. The effect of atmospheric attenuation is also applied to several of the input data cubes. The input object data cubes, $O(\lambda, u, v)$, are used as the input to the CTIS model which produces the corresponding CTIS detector images. Additional photon noise is also applied to several of the detector images. The detector images provide the input to the reconstruction algorithms which provide an estimate, $\widehat{O}(\lambda, u, v)$, of the original input hyperspectral data cube. The chapter concludes with Table 4.4 which summarizes the test cases.

4.1 Binary Star Pair Test

This test generates a hyperspectral cube consisting of a distant binary star pair with each star having a different temperature. The cube is 20 × 20 pixels in the spatial dimensions and contains 15 wavelength bins. The parameters of the stars used are shown in Table 4.1.

Table 4.1: Star Model Parameters

Component	**Model Parameter**	**Value**
Star 1	Temperature	10000^oK
	Radius	2.5 solar radii
	Distance	50 light years
Star 2	Temperature	5000^oK
	Radius	1.1 solar radii
	Distance	50 light years

Note that $1\ solar\ radii = 6.96 \times 10^8 m$ and the model simulates a binary star pair consisting of an A type and G type star as defined in [5]. Each star is treated as a blackbody and using radiometry detailed by Dereniak and Boreman in [7] the number of photons, or photon flux (Φ_p) in $\frac{photons}{sec}$, at the detector can be determined. By selecting a detector integration time of $\Delta t = 0.001 sec$ the binary star data cube is generated as shown in Figure 4.1. The composition of the data cube is also shown in Figure 4.2 where the sum of photons in each wavelength bin for each star is graphed. The resulting CTIS image and x vector sums are shown in Figures 4.3 and 4.4 respectively. The resulting CTIS image and x vector sums are also shown in Figures 4.5 and 4.6 respectively for when the effects of atmospheric attenuation is implemented.

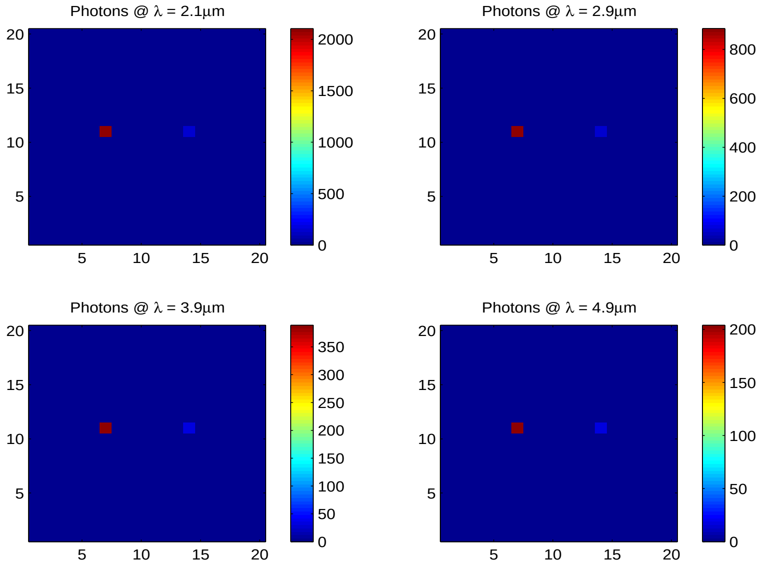


Figure 4.1: This figure shows four wavelength bins of the binary star hyperspectral data cube. The cube has spatial extent of 20 × 20 pixels with the stars being separated by six pixels. The number of photons in each bin is indicated by the color bar.

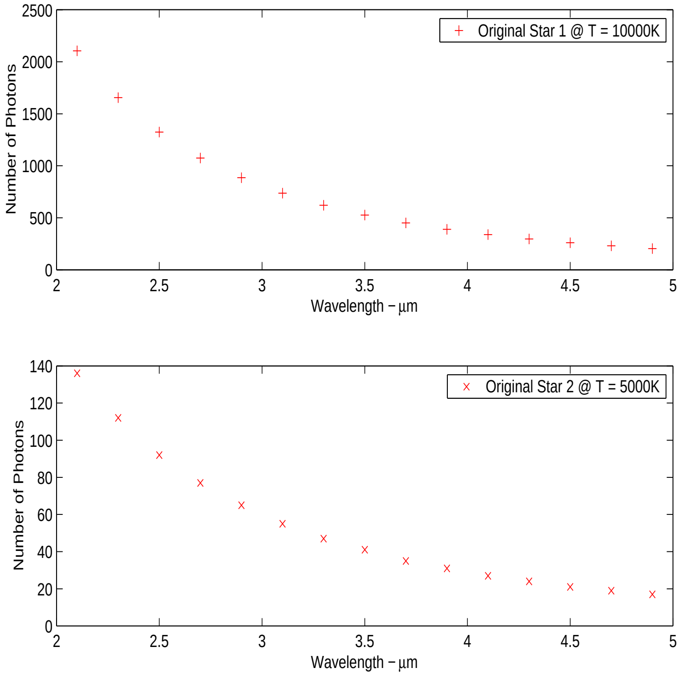


Figure 4.2: This figure shows another view of the binary star hyperspectral cube derived by summing the photons in each bin for each star. Note that both plots exhibit the Planckian shape of the modelled blackbody stars.

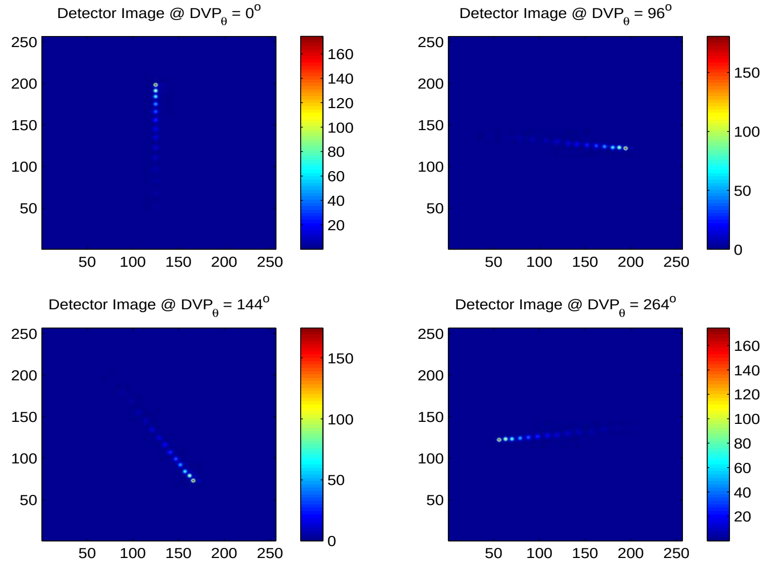


Figure 4.3: This figure shows the detector image at the four DVP rotation angles resulting from the CTIS sensing the binary star hyperspectral data cube when no atmospheric attenuation is considered.

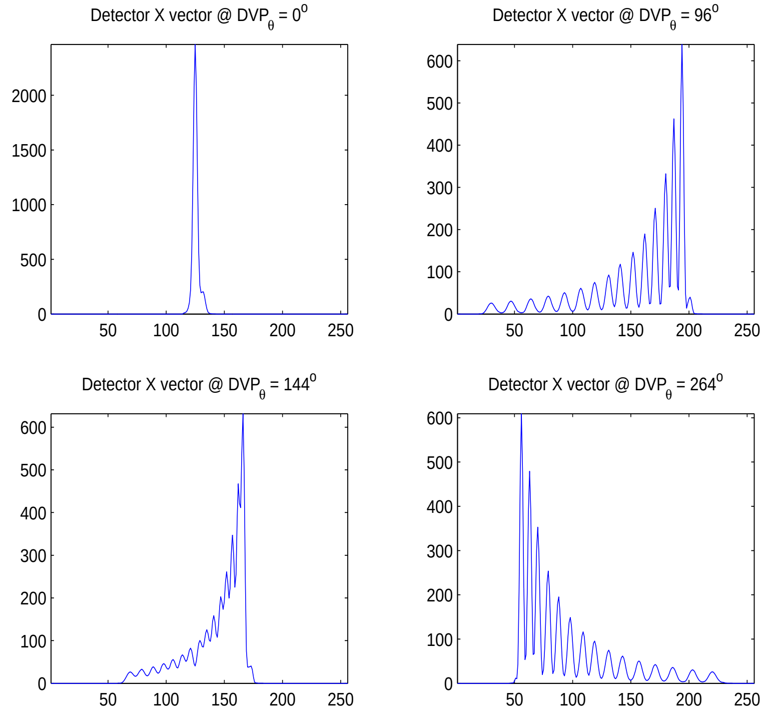


Figure 4.4: This figure shows the x vector sum of the detector image for the binary star data cube. No atmospheric attenuation is implemented in determining these detector vectors.

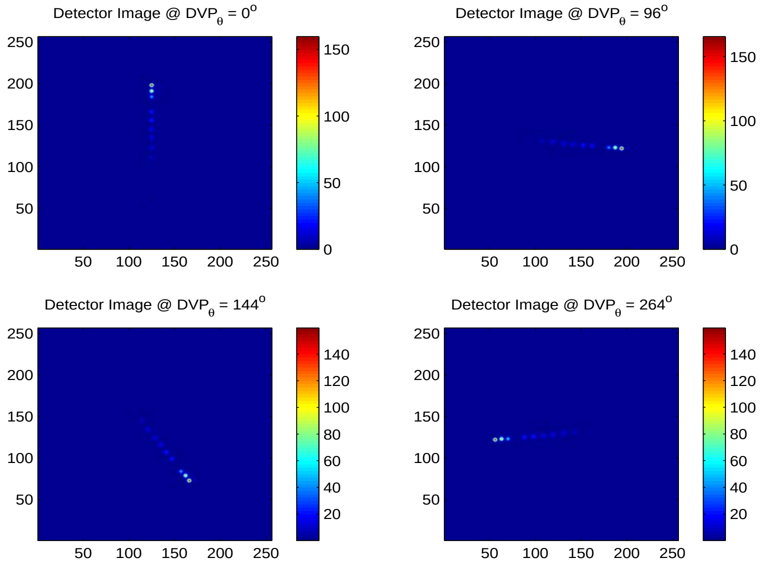


Figure 4.5: This figure shows the detector image at the four DVP rotation angles resulting from the CTIS sensing the binary star hyperspectral data cube when the effects of atmospheric attenuation is applied.

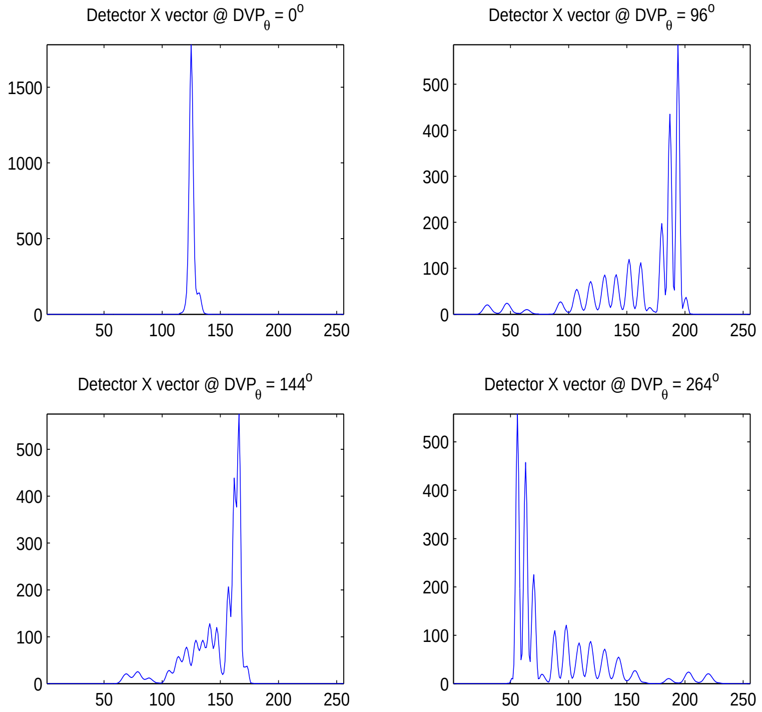


Figure 4.6: This figure shows the x vector sum of the detector image for the binary star data cube. These detector vectors implement atmospheric attenuation.

4.2 Spatially Separate Monochromatic Source Test

The binary star test case in section 4.1 contains an input hyperspectral scene with limited spatial information. A follow-on class of test cases construct hyperspectral scenes with both spectral and spatial features. A similar hyperspectral scene test case to that used by Gustke in chapter 3 of [11] is generated. This test uses five spatially separate monochromatic extended sources in five different wavelength bins. Thus the source "bars" are also spectrally separated. The composition of the bar test hyperspectral data cube is shown in Figure 4.7 where the spatial extent of the scene is 100 × 100 pixels. The resulting CTIS detector image is shown in Figure 4.8. Figure 4.9 also shows the detector image with the addition of random noise.

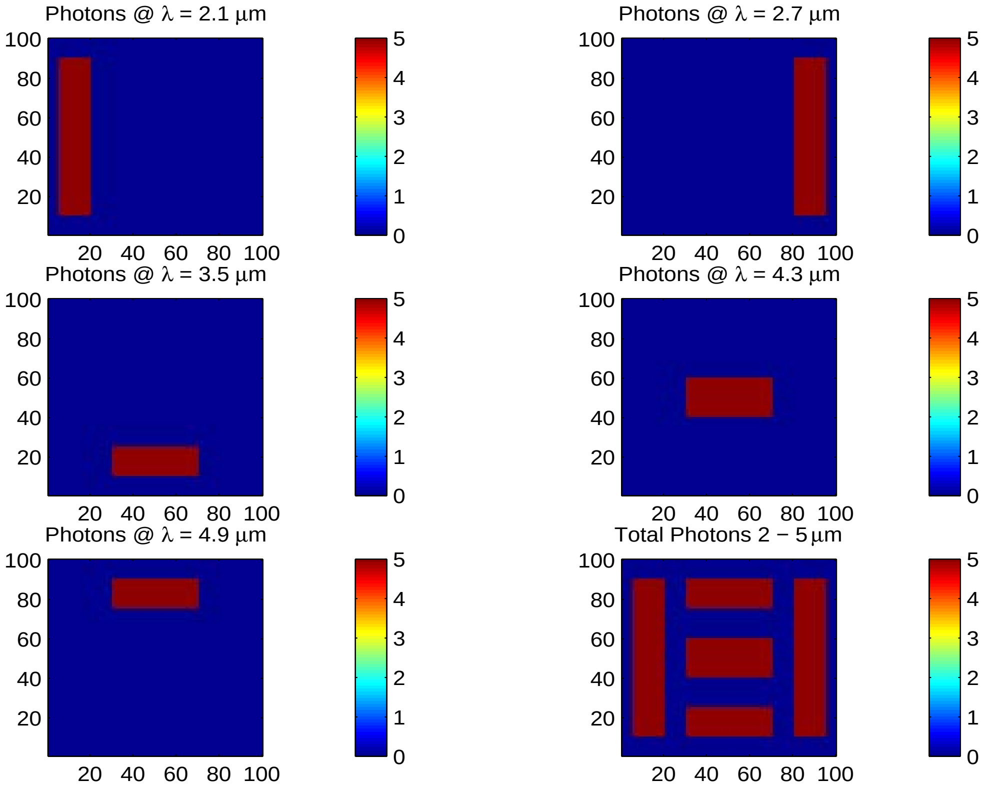


Figure 4.7: The bar test consists of five monochromatic sources in five spectral bins. The sources have a uniform five photons across their extent. The bins that are not shown contain zero photons. The final subplot shows a total $2 - 5\mu m$ view of the source data.

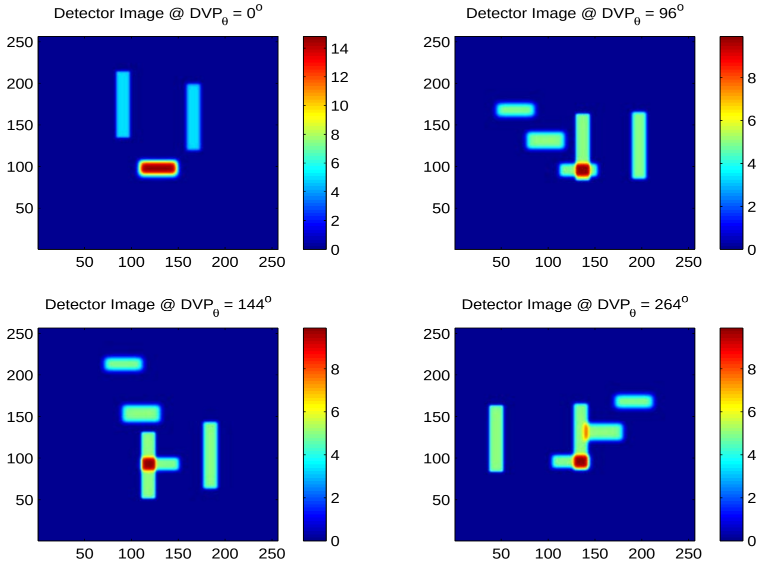


Figure 4.8: This figure shows the detector image at four rotation angles resulting from the CTIS model sensing the 100 × 100 pixel bar test hyperspectral data cube. The CTIS model spatially disperses the sources as a function of wavelength and DVP rotation angle.

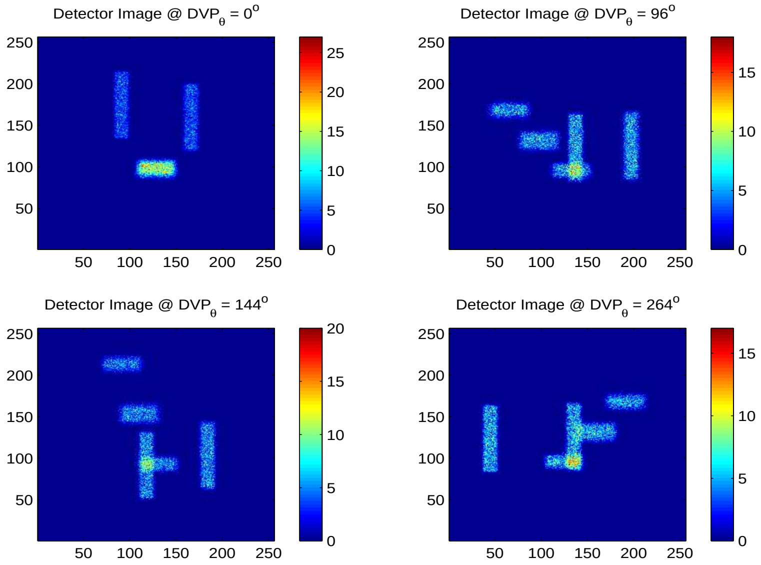


Figure 4.9: This figure shows the addition of Poisson random noise to the CTIS detector image of the bar test hyperspectral data cube.

A smaller bar test hyperspectral data cube with identical spatial and spectral features to Figure 4.7 is also generated on a 20×20 pixel spatial scene. This data cube is used to show the utility of the vector reconstruction algorithm when reconstructing smaller spatial scenes. The small bar test data cube is used with the $66.67\mu m$ pixel pitch detector with the resulting noiseless CTIS detector image shown in Figure 4.10 and noiseless image x vector shown in Figure 4.11.

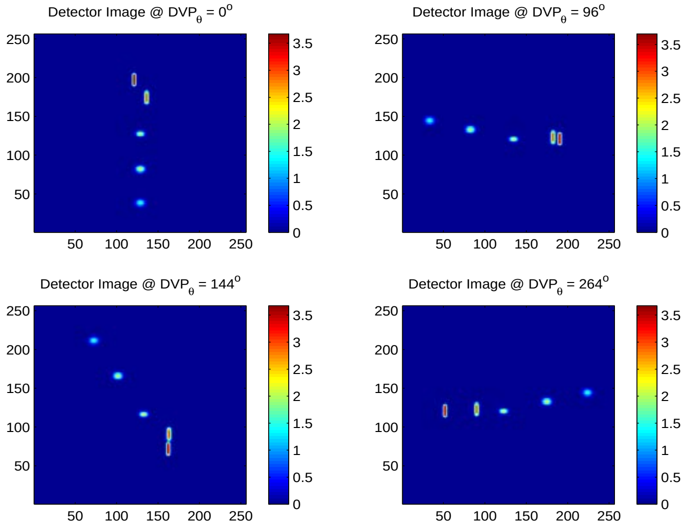


Figure 4.10: This figure shows the detector at four rotation angles resulting from the CTIS model sensing the 20×20 pixel bar test hyperspectral data cube. The major difference (apart from bar size) compared to Figure 4.8 is that the spectrally dispersed bars are spatially separated. Note also how the expected bar-shape changes. This is a function of the PSF of the optics in that PSF width increases with wavelength.

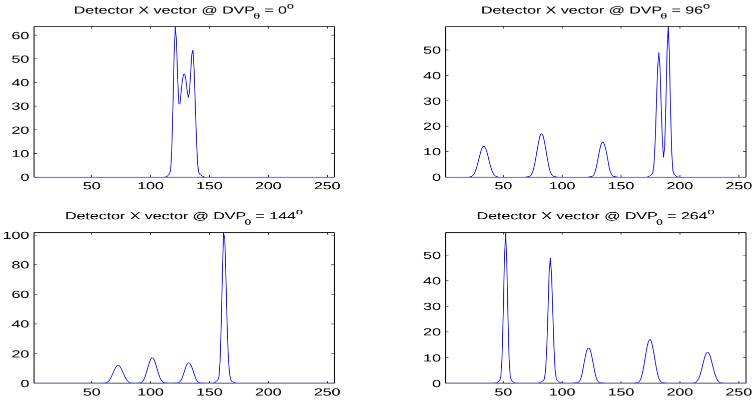


Figure 4.11: This figure shows the x vector sum of the detector image for the small bar test hyperspectral data cube. The vector again shows the spatial separation of the dispersed bars.

With the addition of noise the resulting CTIS detector image and x vector sum are shown in Figures 4.12 and 4.13 respectively.

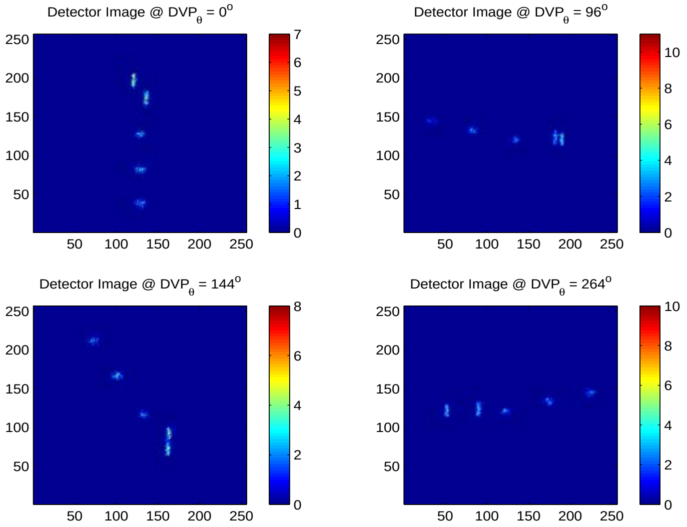


Figure 4.12: This figure shows the addition of random noise to the CTIS image of the 20 × 20 pixel bar test hyperspectral data cube.

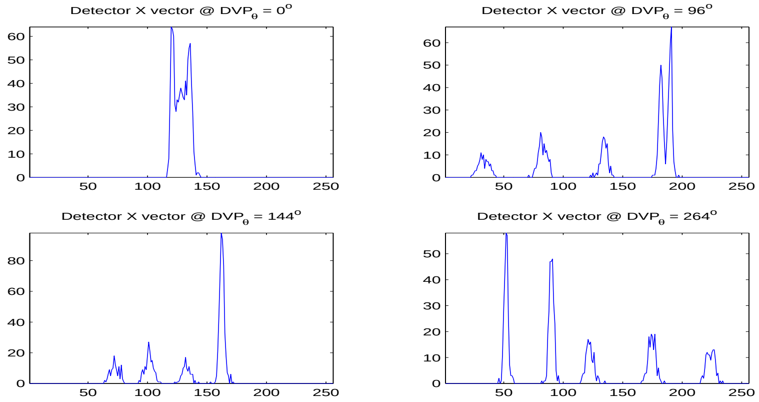


Figure 4.13: This figure shows the x vector sum of the noisy detector image for the small bar test hyperspectral data cube.

4.3 Spatially Overlapping Monochromatic Source Test

This test case uses monochromatic source bars which spatially overlap within the hyperspectral data cube. The overlapping bar test cube with a spatial extent of 100×100 pixels is generated using source bars in four spectral bins as shown in Figure 4.14. The resulting CTIS detector image is shown in Figure 4.15. Figure 4.16 also shows the detector image with the addition of random noise.

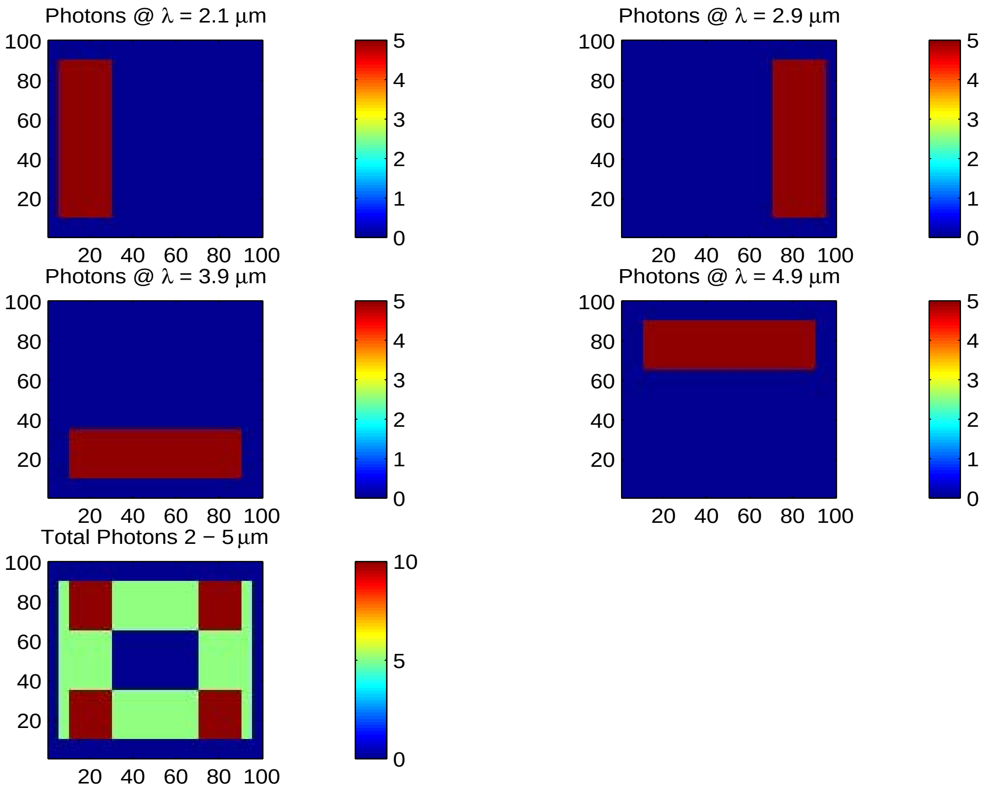


Figure 4.14: The overlapping bar test consists of four monochromatic sources in four spectral bins. The sources have a uniform five photons across their extent. The bins that are not shown contain zero photons. The final subplot shows a total $2 - 5\mu m$ view of the source data.

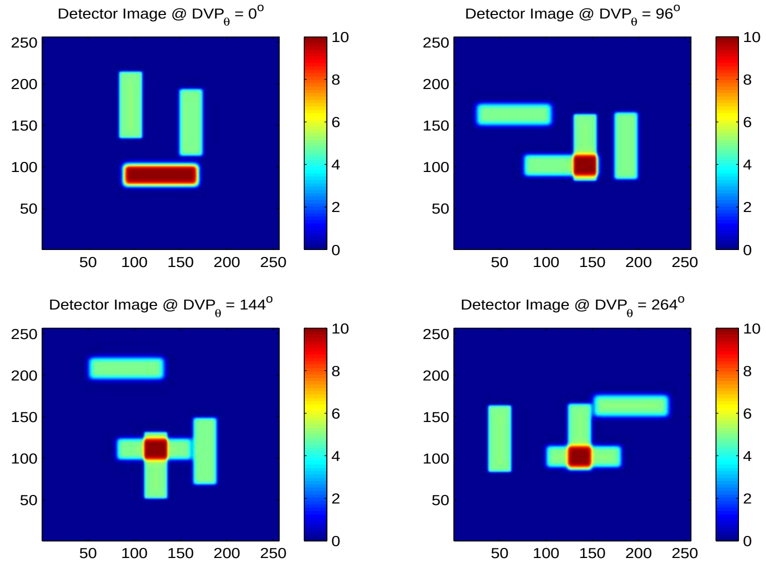


Figure 4.15: This figure shows the detector image at four rotation angles resulting from the CTIS model sensing the 100 × 100 pixel overlapping bar test hyperspectral data cube.

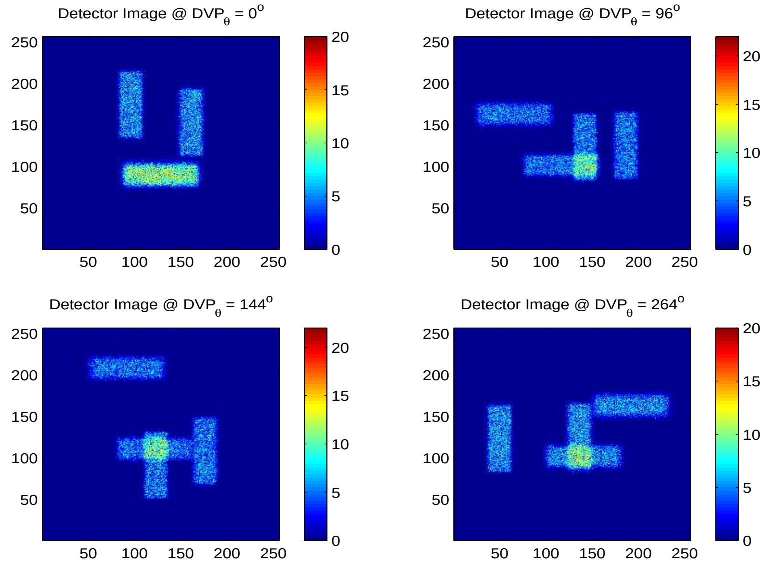


Figure 4.16: This figure shows the addition of Poisson random noise to the CTIS detector image of the 100 × 100 pixel overlapping bar test hyperspectral data cube.

A smaller overlapping bar test hyperspectral data cube is also generated on a 20×20 pixel spatial scene. The small overlapping bar test data cube is used with the $66.67\mu m$ pixel pitch detector with the resulting noiseless CTIS detector image shown in Figure 4.17 and noiseless image x vector shown in Figure 4.18.

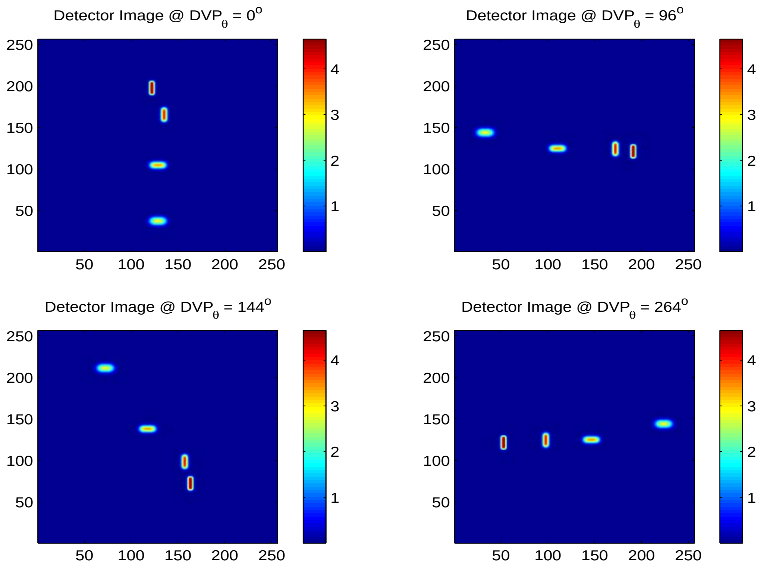


Figure 4.17: This figure shows the detector at four rotation angles resulting from the CTIS model sensing the 20×20 pixel bar test hyperspectral data cube. Again the spectrally dispersed bars are spatially separated with the smaller scene image.

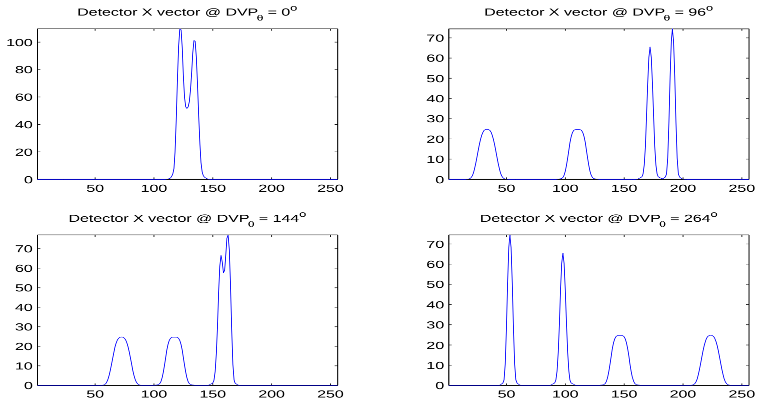


Figure 4.18: This figure shows the x vector sum of the detector image for the small overlapping bar test hyperspectral data cube. The vector again shows the spatial separation of the dispersed bars.

With the addition of noise the resulting CTIS detector image and x vector sum are shown in Figures 4.19 and 4.20 respectively.

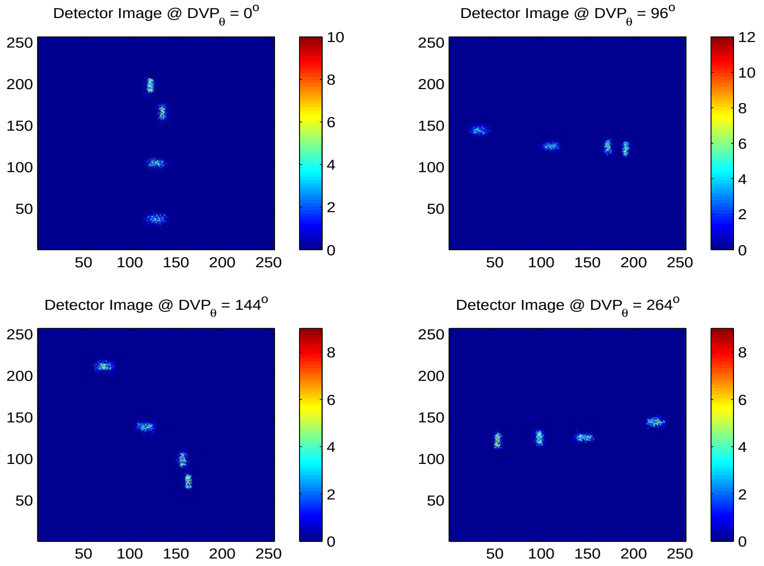


Figure 4.19: This figure shows the addition of random noise to the CTIS image of the 20 × 20 pixel overlapping bar test hyperspectral data cube.

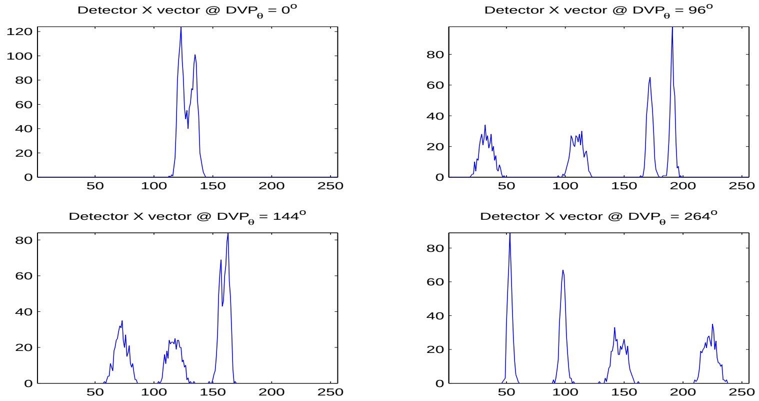


Figure 4.20: This figure shows the x vector sum of the noisy detector image for the small overlapping bar test hyperspectral data cube.

4.4 Monochromatic Numbers Source Test

The final monochromatic source test consists of a series of numbers in consecutive wavelength bins as shown in Figure 4.21. The lack of uniformity in the handwritten numbers provides minimal spatial correlation between consecutive wavelength bins. Hence this should provide a “worst case” scenario increasing the challenge provided to the reconstruction algorithm.

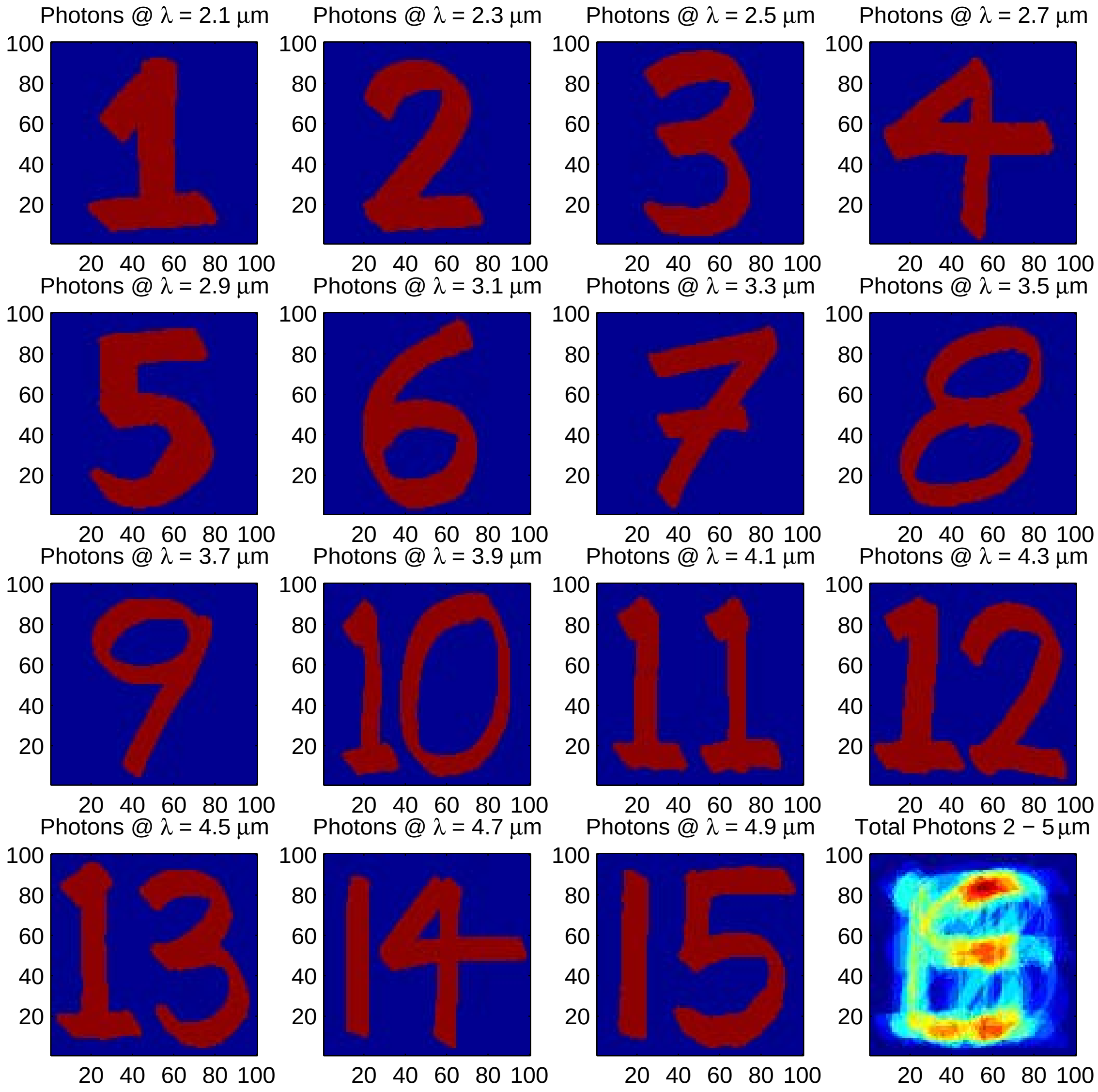


Figure 4.21: The numbers test consists of fifteen monochromatic sources in the different spectral bins. The sources have a uniform one hundred photons across their extent. The final subplot shows a total $2 - 5\mu m$ view of the source data.

The noiseless CTIS detector image resulting from the numbers test data cube is shown in Figure 4.22. Similarly, the noisy detector image and atmospherically attenuated detector image are shown in Figures 4.23 and 4.24 respectively.

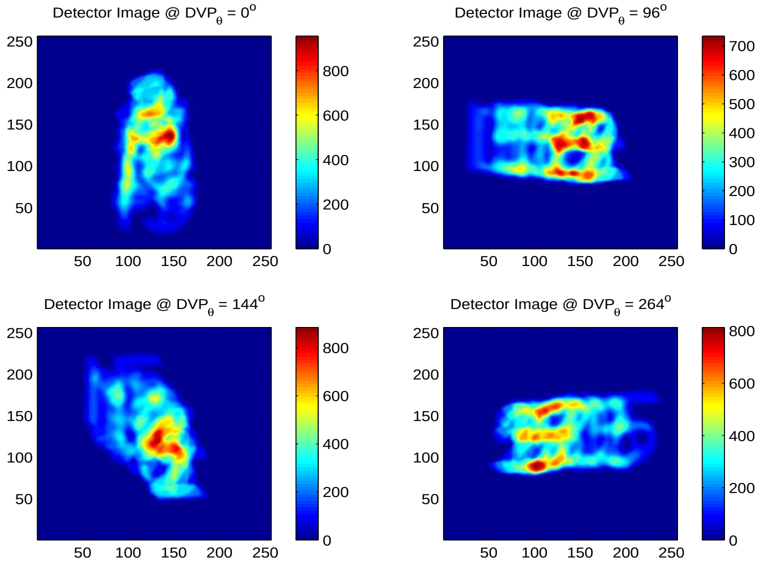


Figure 4.22: This figure shows the detector at four rotation angles resulting from the CTIS model sensing the numbers test hyperspectral data cube.

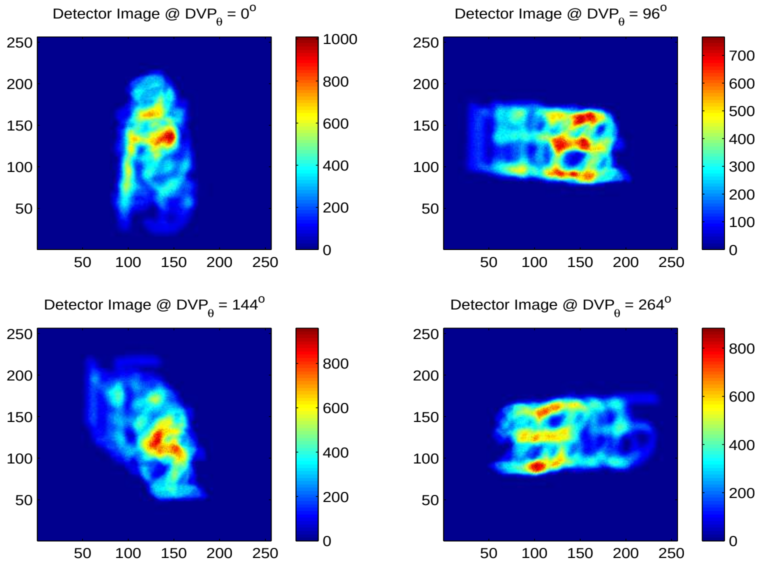


Figure 4.23: This figure shows a noisy detector image resulting from the CTIS model sensing the numbers test hyperspectral data cube.

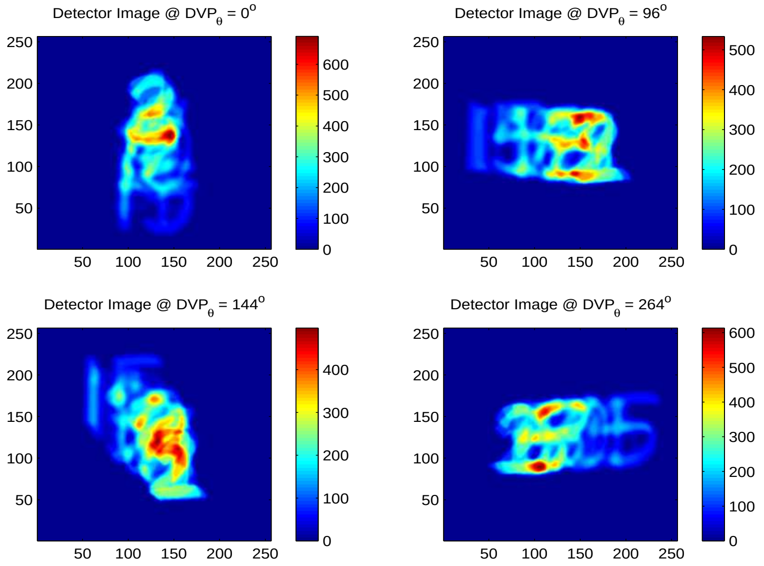


Figure 4.24: This figure shows the CTIS detector image of the numbers test hyperspectral data cube after the spectral data has been attenuated by the atmosphere.

4.5 Static Fireball Tests

4.5.1 Uniform Fireball Test. This test case simulates a uniform temperature fireball appearing on a random temperature background scene within the CTIS FOV. The scenario is based on data collected by AFIT and documented by Dills and Perram in [8]. Three different temperature fireballs are generated with the main parameters shown in Table 4.2.

Table 4.2: Uniform Fireball Model Parameters

Component	**Model Parameter**	**Value**
Fireball	Temperature 1	400^oK
	Temperature 2	1000^oK
	Temperature 3	1600^oK
	Surface Area (sphere)	$4000m^2$
	Distance	$3258m$
Background	Mean Temperature	300^oK

At the given range, the fire ball diameter extends approximately one tenth of the FOV of the forward lens in the FOS as shown in Figure 4.29. The background temperature is generated as a normal random variable with mean 300^oK and standard deviation 10^oK.

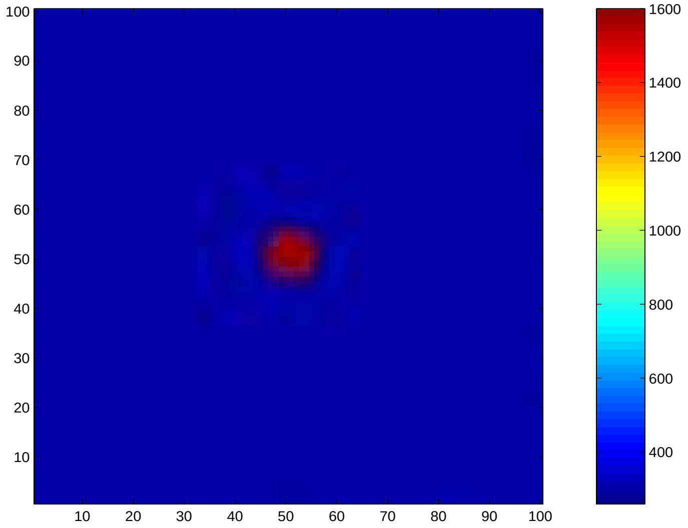


Figure 4.25: This figure shows a temperature map of the 1600^oK fireball located in the center of a random background scene with a mean temperature of 300^oK .

Each point in the scene is modelled as blackbody with the resulting total number of photons in each wavelength bin shown in Figure 4.26. This figure also shows the effect of atmospheric attenuation on the photon sum in each wavelength bin.

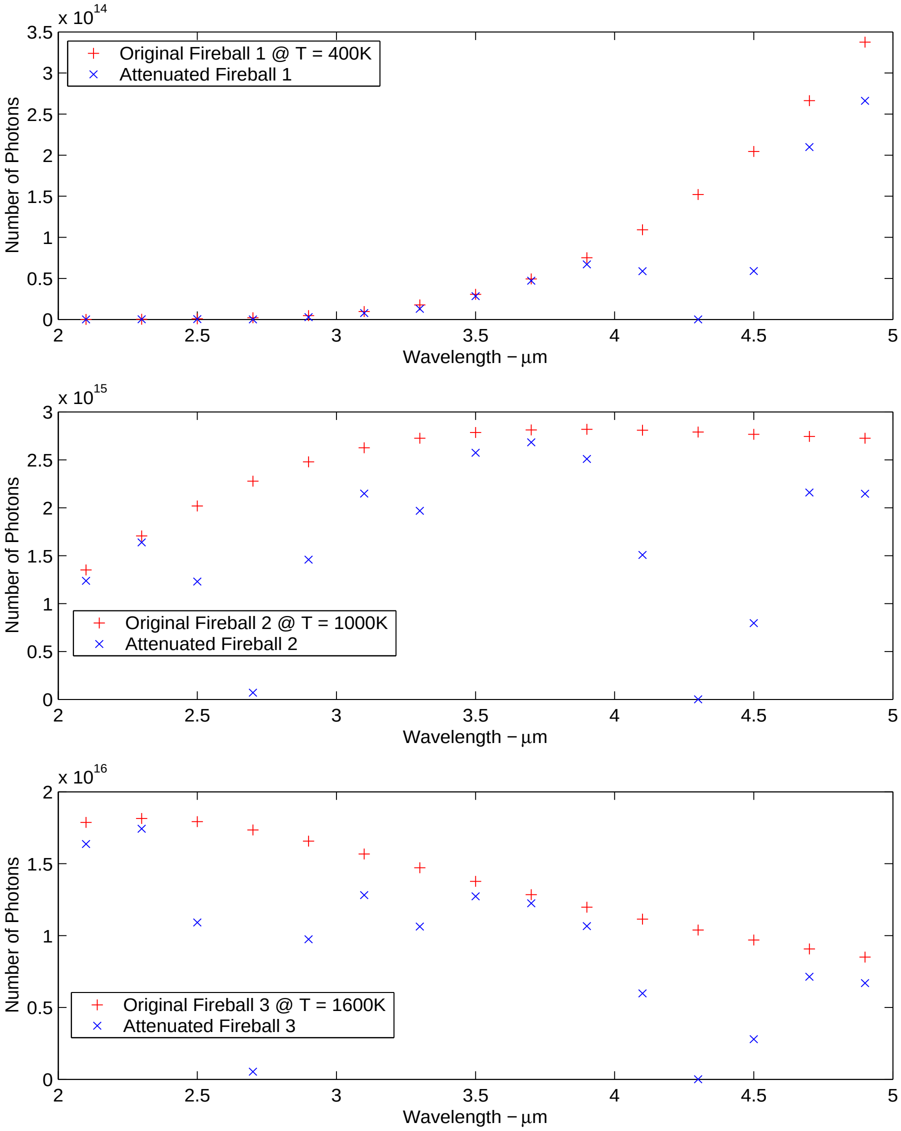


Figure 4.26: This figure shows the number of photons in each wavelength bin of the hyperspectral data cube for fireballs at 400^{o}K, 1000^{o}K and 1600^{o}K on a 300^{o}K background. The figure also shows the effects of atmospheric attenuation on the number of photons arriving at the detector.

The CTIS detector image and x vector sum for the 400^{o}K, 1000^{o}K and 1600^{o}K fireballs are shown in Figures 4.27, 4.28 and 4.29 respectively.

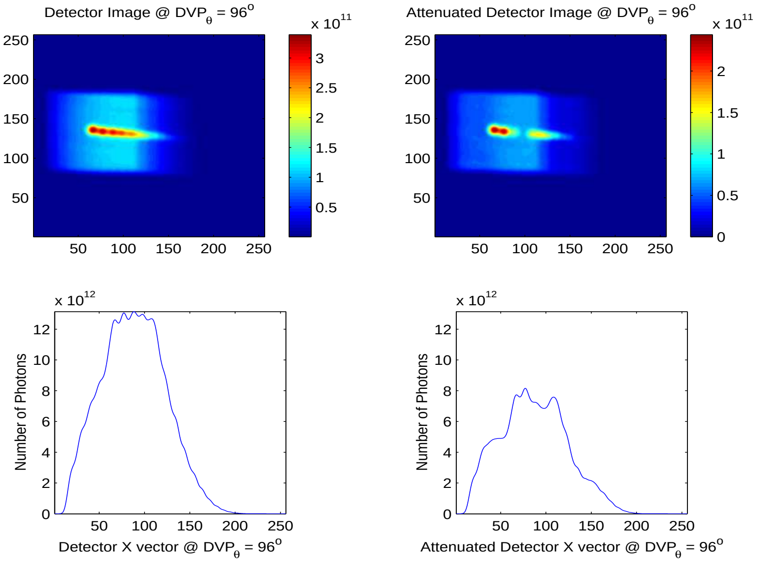


Figure 4.27: This figure shows the CTIS detector image and x vector sum of the 400^{o}K fireball for both no atmosphere and with atmospheric attenuation.

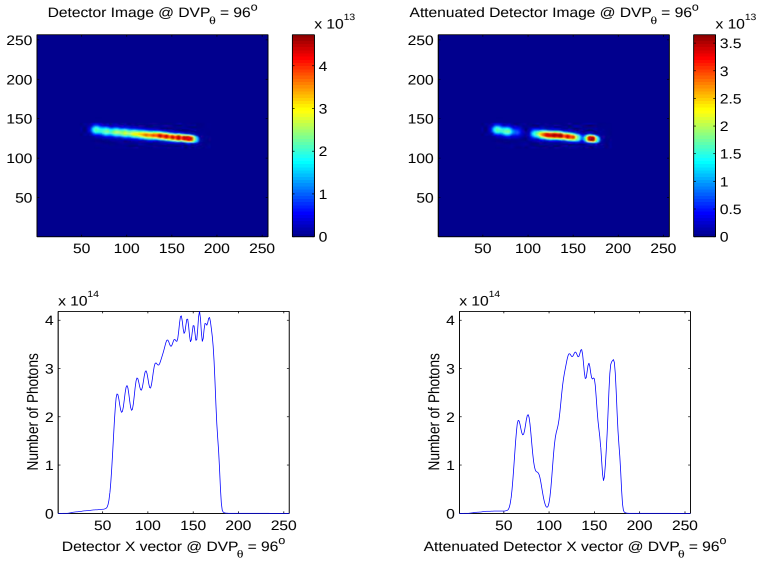


Figure 4.28: This figure shows the CTIS detector image and x vector sum of the 1000^{o}K fireball for both no atmosphere and with atmospheric attenuation.

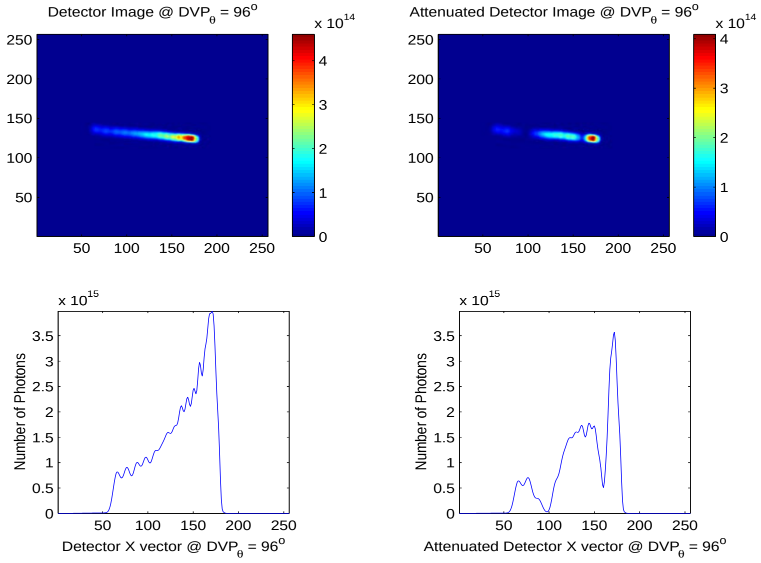


Figure 4.29: This figure shows the CTIS detector image and x vector sum of the 1600^{o}K fireball for both no atmosphere and with atmospheric attenuation.

4.5.2 Concentric Temperature Rings Fireball Test. This test case simulates a variable temperature fireball appearing on a random temperature background scene within the CTIS FOV. The fireball consists of constant temperature concentric rings at 800^{o}K, 1000^{o}K, 1200^{o}K, 1500^{o}K and 1600^{o}K. The background temperature is a normal random variable with a mean temperature of 300^{o}K and a standard deviation of 10^{o}K. The diameter of the fireball has been enlarged to extended one half of the FOV as shown is Figure 4.30.

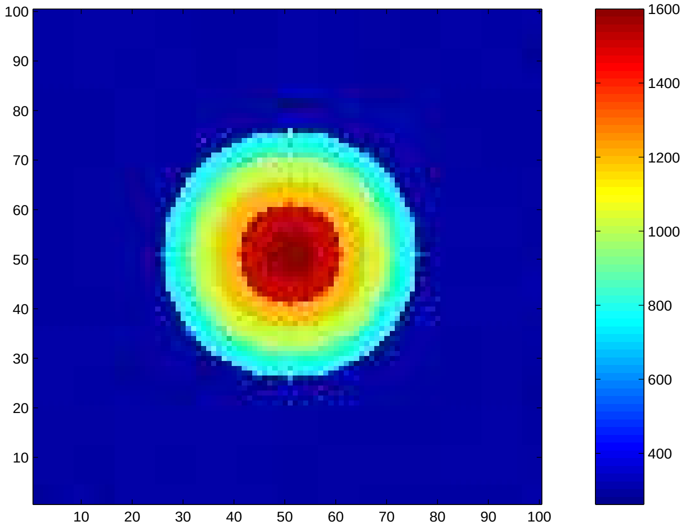


Figure 4.30: This figure shows the temperature map of a fireball consisting of concentric rings of constant temperature at 800^{o}K, 1000^{o}K, 1200^{o}K, 1500^{o}K and 1600^{o}K located in the center of a random background scene with a mean temperature of 300^{o}K.

The photons in four of the wavelength bins resulting from the temperature scene in Figure 4.30 is shown in Figure 4.31. The clean and atmospherically attenuated CTIS detector image and x vector sum for the concentric rings fireball are shown Figure 4.32.

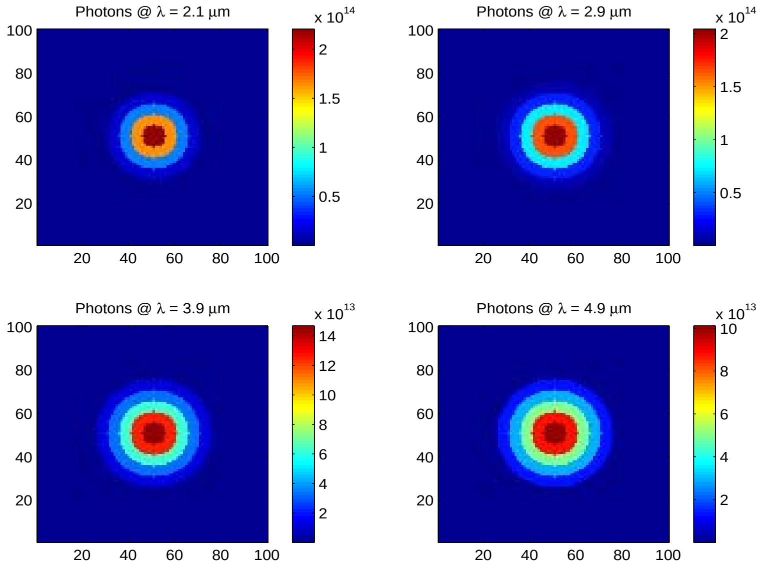


Figure 4.31: This figure shows the composition of photons in four wavelength bins resulting from the concentric rings fireball. The rings of constant temperature are at 800^{o}K, 1000^{o}K, 1200^{o}K, 1500^{o}K and 1600^{o}K on the 300^{o}K random background scene.

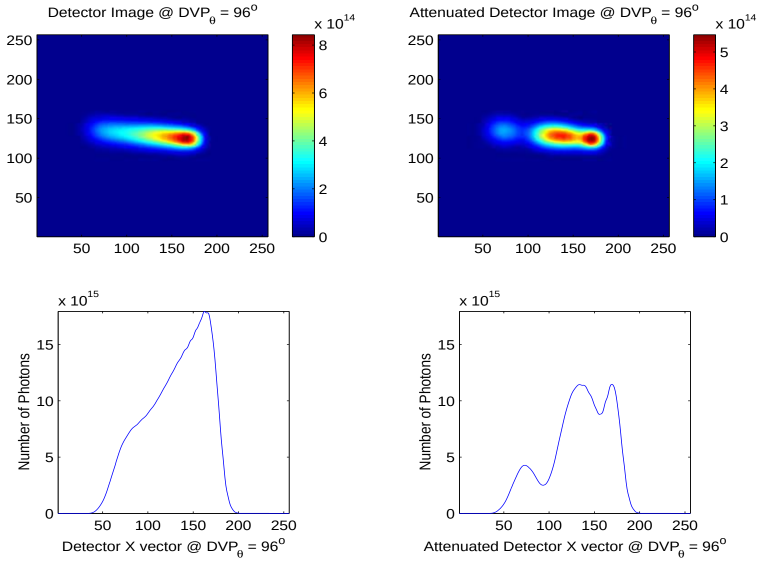


Figure 4.32: This figure shows the CTIS detector image and x vector sum of the concentric rings fireball for both no atmosphere and with atmospheric attenuation.

4.5.3 Hot-Spot Fireball Test. This test case simulates a fireball containing several different temperature hot-spots appearing on a random temperature background scene within the CTIS FOV. The hot-spots in the fireball are at at 1000^oK, 1200^oK, 1500^oK and 1600^oK while the remainder of the fireball is at 600^oK as shown in Figure 4.33. The background temperature is a normal random variable with a mean of 300^oK.

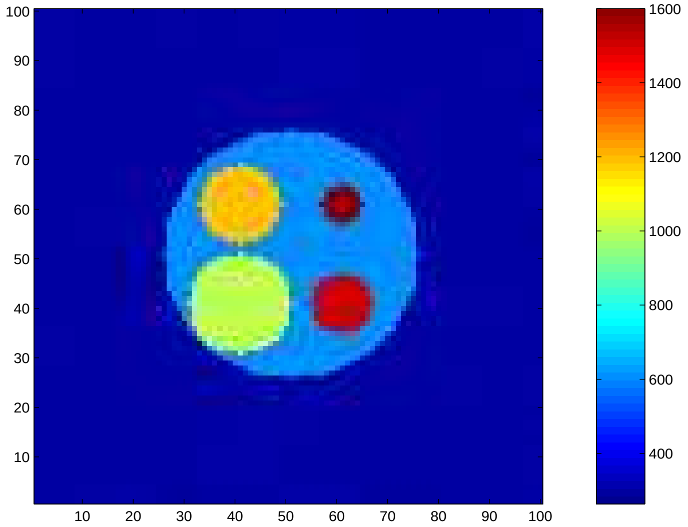


Figure 4.33: This figure shows a temperature map of a 600^oK fireball containing hot-spots at a temperature of 1000^oK, 1200^oK, 1500^oK and 1600^oK. The random background scene has a mean temperature of 300^oK.

The photons in four of the wavelength bins resulting from the temperature scene in Figure 4.33 is shown in Figure 4.34. The clean and atmospherically attenuated CTIS detector image and x vector sum for the hot-spots fireball are shown Figure 4.35.

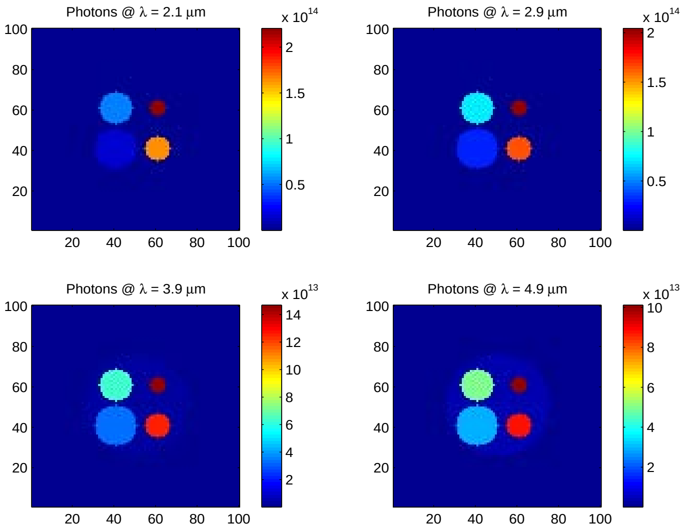


Figure 4.34: This figure shows the composition of photons in four wavelength bins resulting from the hot-spots fireball. The temperature of the hot-spots are at 1000^{o}K, 1200^{o}K, 1500^{o}K and 1600^{o}K while the remainder of the fireball is at 600^{o}K. The random background temperature of the scene is at a mean temperature of 300^{o}K.

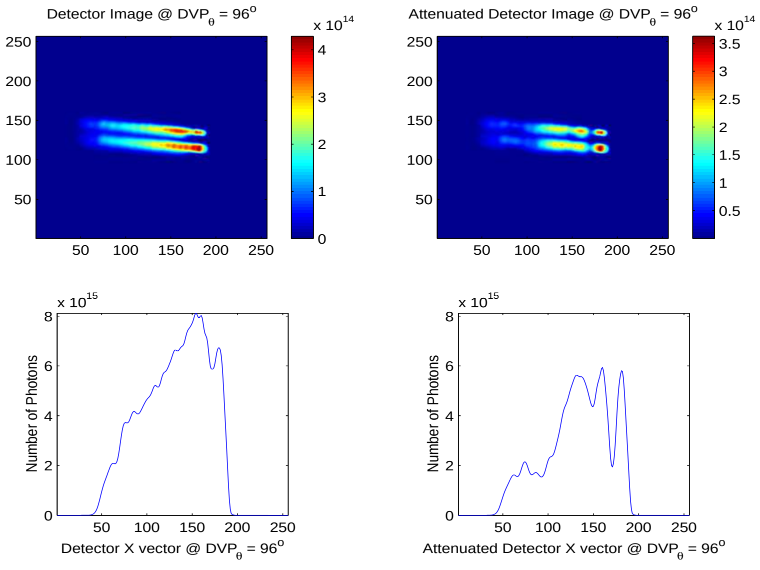


Figure 4.35: This figure shows the CTIS detector image and x vector sum of the hot-spots fireball for both no atmosphere and with atmospheric attenuation.

4.6 Evolving Fireball Test

This test case simulates a temporally evolving, uniform temperature fireball appearing on a random temperature background scene within the CTIS FOV. This is achieved by combining the photon flux data of different uniform fireballs in section 4.5.1 so that the photon flux at the DVP is not constant throughout one complete revolution of the DVP. The four test cases in this section consist of evolving fireballs with parameters as shown in Table 4.3 where the fireballs with two temperatures transition at $DVP_\theta = 168^o$ and the fireballs with three temperatures transition at $DVP_\theta = 120^o$ and $DVP_\theta = 240^o$. The CTIS detector image and x vector sum for the four evolving fireballs are shown in Figures 4.36, 4.37, 4.38 and 4.39 respectively.

Table 4.3: Evolving Fireball Model Parameters

Component	**Model Parameter**	**Value**
Fireball 1	Temperature 1 Temperature 2	1600^{o}K 1500^{o}K
Fireball 2	Temperature 1 Temperature 2	1600^{o}K 1300^{o}K
Fireball 3	Temperature 1 Temperature 2 Temperature 3	1600^{o}K 1400^{o}K 1200^{o}K
Fireball 4	Temperature 1 Temperature 2 Temperature 3	1600^{o}K 1000^{o}K 400^{o}K
All Fireballs	Surface Area (sphere) Distance	4000m^2 3258m
Background	Mean Temperature	300^{o}K

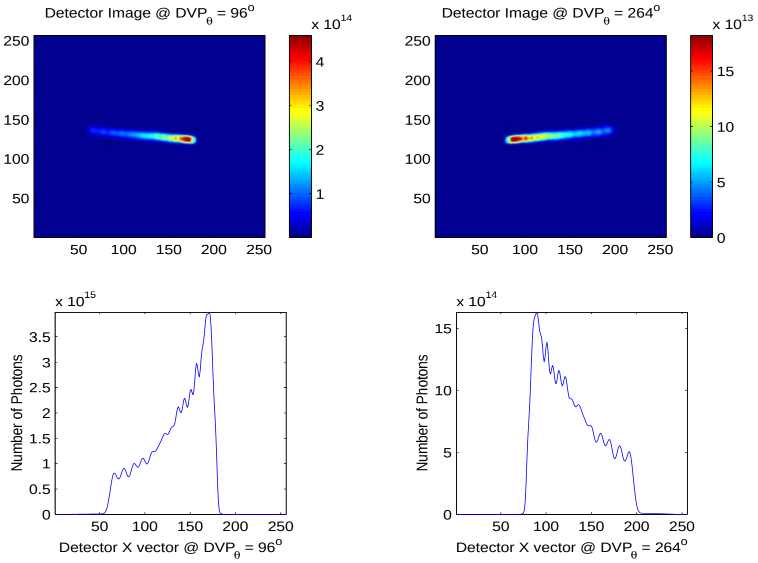


Figure 4.36: This figure shows the CTIS detector image and x vector sum of the first evolving fireball. This fireball transitions from 1600^{o}K to 1500^{o}K at $DVP_\theta = 168^o$.

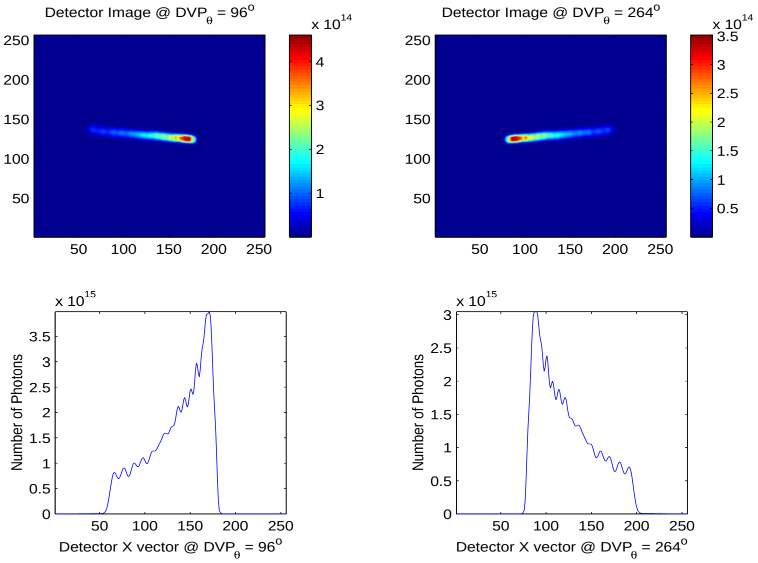


Figure 4.37: This figure shows the CTIS detector image and x vector sum of the second evolving fireball. This fireball transitions from 1600^{o}K to 1300^{o}K at $DVP_\theta = 168^o$.

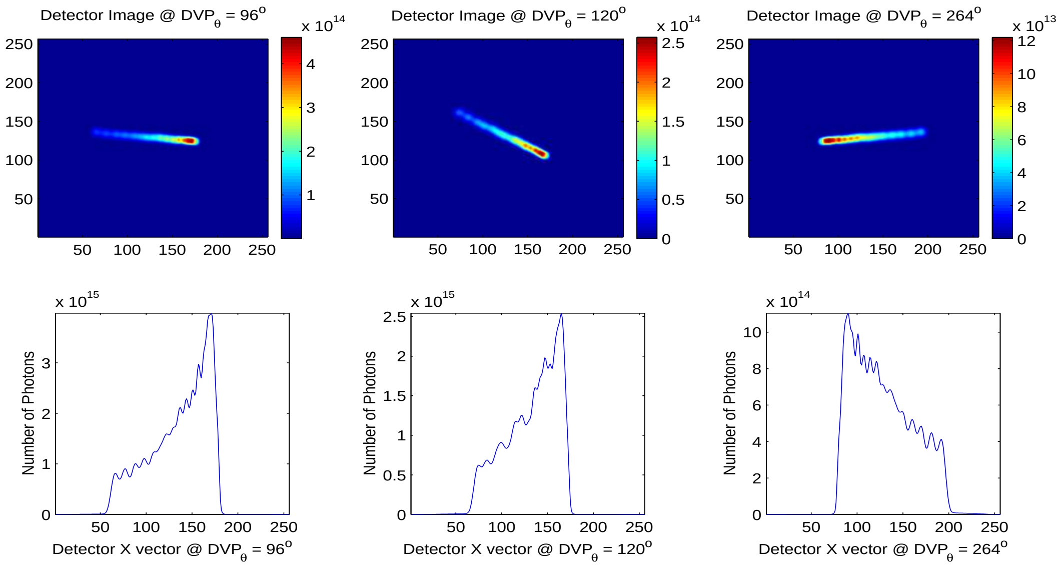


Figure 4.38: This figure shows the CTIS detector image and x vector sum of the third evolving fireball. This fireball transitions from 1600^{o}K to 1400^{o}K at $DVP_\theta = 120^o$ and 1400^{o}K to 1200^{o}K at $DVP_\theta = 240^o$.

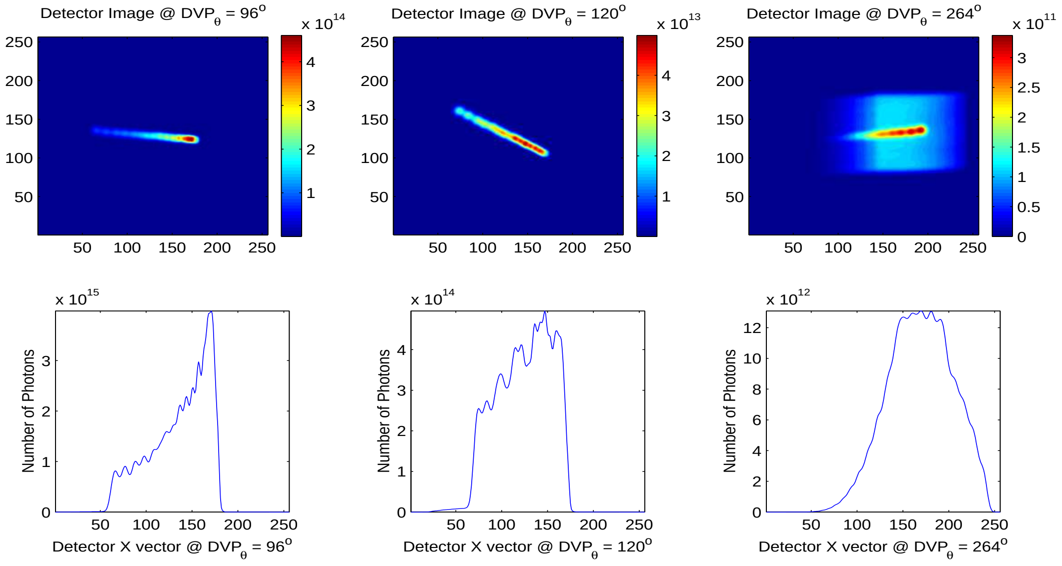


Figure 4.39: This figure shows the CTIS detector image and x vector sum of the fourth evolving fireball. This fireball transitions from 1600^{o}K to 1000^{o}K at $DVP_\theta = 120^o$ and 1000^{o}K to 400^{o}K at $DVP_\theta = 240^o$.

4.7 Summary of Test Cases

The test case scenarios developed in this chapter are labelled and summarized in Table 4.4.

Table 4.4: Test Case Summary

Test Scenario	**Section**	**Test**	**Object Size**	**Atmosphere**	**Noise**	**Reconstruction**
Binary Star Pair	4.1	1	20×20	No	No	2D
		2	20×20	Yes	No	2D + atm
		3	20×20	No	No	1D
		4	20×20	Yes	No	1D + atm
Spatially Separate Monochromatic Source	4.2	1	100×100	No	No	2D
		2	100×100	No	Yes	2D
		3	100×100	No	No	1D
		4	100×100	No	Yes	1D
		5	20×20	No	No	1D
		6	20×20	No	Yes	1D
Spatially Separate Monochromatic Source	4.3	1	100×100	No	No	2D
		2	100×100	No	Yes	2D
		3	100×100	No	No	1D
		4	100×100	No	Yes	1D
		5	20×20	No	No	1D
		6	20×20	No	Yes	1D
Monochromatic Numbers Source	4.4	1	100×100	No	No	2D
		2	100×100	Yes	No	2D + atm
		3	100×100	No	Yes	2D
Uniform Fireball 400^oK, 1000^oK, & 1600^oK	4.5.1	1	100×100	No	No	1D
		2	100×100	Yes	No	1D + atm
		3	100×100	No	No	1D
		4	100×100	Yes	No	1D + atm
		5	100×100	No	No	1D
		6	100×100	Yes	No	1D + atm
Concentric Temperature Rings Fireball	4.5.2	1	100×100	No	No	2D
		2	100×100	Yes	No	2D + atm
		3	100×100	No	No	1D
		4	100×100	Yes	No	1D + atm
Hot-Spot Fireball	4.5.3	1	100×100	No	No	2D
		2	100×100	Yes	No	2D + atm
		3	100×100	No	No	1D
		4	100×100	Yes	No	1D + atm
Evolving Fireball	4.6	1	100×100	No	No	1D
		2	100×100	No	No	1D
		3	100×100	No	No	1D
		4	100×100	No	No	1D

V. Results and Analysis of Reconstruction Performance

This chapter demonstrates the performance of the reconstruction algorithms in section 3.4 in estimating the input hyperspectral data cube for each test case in chapter IV. Reconstruction is performed on both the 2D detector image and the 1D x vector sum for several of the test cases. The treatment of atmospheric attenuation and the addition of noisy detector data is also demonstrated in the test cases. The test case results are presented in the same order as chapter IV and are labelled as indicated in Table 4.4. The chapter concludes with an analysis of the performance of the 2D and vector reconstruction algorithms.

5.1 Metrics for Reconstruction Performance

The metrics used for evaluating and comparing the reconstruction performance achieved in the test cases include:

5.1.1 Photon Sum. This metric compares the number of photons in each wavelength for both the original and reconstructed hyperspectral data cube. The results tables give Photon Sum for both data cubes in units of *photons* and also as a percentage with the original data cube taken as the reference.

5.1.2 Reconstructed Temperature Estimate. The reconstructed photon numbers are also used to estimate a reconstructed temperature for the test cases consisting of blackbody sources. This is achieved by fitting the Planckian expression to a spectral sample obtained at one pixel location, ie, the number of photons in each wavelength bin for one spatial location. This is directly applicable to the 2D reconstructed data cubes. However the photon numbers from data cubes resulting from vector reconstructions must be tailored by considering the column sum composition of the imaged scene. For example the uniform fireballs span one tenth of a linear dimension of the detector. Thus the number of photons must be scaled depending on the location of the pixel where the temperature is required. The scaling ranges from dividing by ten if viewing the fireball center, to unit scaling if viewing the fireball edge. This also works better where the fireball temperature is significantly larger than the background temperature. This scenario causes the photons numbers arriving at the detector from the fireball to be well above the photons arriving from the background. An estimated temperature is provided in the summary of each of the blackbody test case scenarios.

5.1.3 Reconstruction Error Metric.

The Reconstruction Error Metric (REM) is a 1-norm measure of the spatial error in each wavelength bin calculated by summing the absolute value of the difference of photons in the reconstructed and original data cube. For 2D reconstructions the REM is mathematically expressed as

$$REM(\lambda) = \sum_{u=1}^{N} \sum_{v=1}^{N} |\widehat{O}(\lambda, u, v) - O(\lambda, u, v)| \tag{5.1}$$

where N is the linear spatial dimension of the data cube. Vector reconstructions reduce the REM to one spatial dimension as

$$REM_u(\lambda) = \sum_{u=1}^{N} |\widehat{O}(\lambda, u) - O(\lambda, u)| \tag{5.2}$$

The REM provides a measure of incorrectly located photons in each wavelength bin of the reconstructed data cube. The absolute value provides the same penalty to both deficient and excessive numbers of photons at each pixel. The results tables show the REM in units of *photons* and as a percentage comparison to the number of photons in the original data cube.

5.1.4 Spectral Bleeding.

In several of the monochromatic source test scenarios some of the wavelength bins in the scene hyperspectral data cube contain zero photons. Non-zero photon numbers in the corresponding reconstructed bins indicates bleeding, or leakage, of photons during the reconstruction. The degree of spectral bleeding is quantified by comparing photons numbers in the nominally empty reconstructed bins with the photon numbers in the original data cube.

5.2 Binary Star Pair Results

The results for test 4.1.1 are obtained by performing 100 iterations of the 2D reconstruction algorithm. Four wavelength bins of the reconstructed input hyperspectral data cube, $\widehat{O}(\lambda, u, v)$, are shown in Figure 5.1. The reconstructed data cube exhibits similar spatial features to the original data cube in Figure 4.1 with the addition of blurring induced by the CTIS optics.

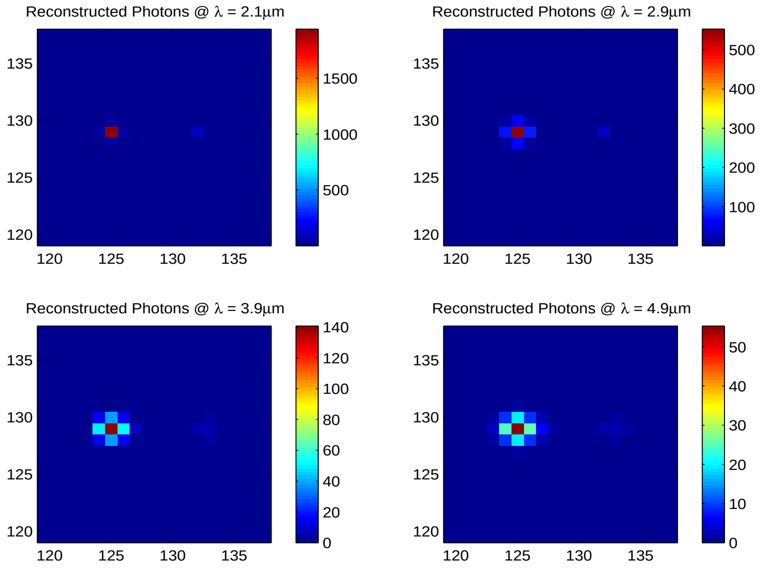


Figure 5.1: This figure shows four wavelength bins of the reconstructed input hyperspectral data cube obtained by performing 100 iterations of the 2D reconstruction algorithm.

The sum of photons in each wavelength bin for each star is shown in Figure 5.2 indicating that 2D reconstruction algorithm is tracking the total number of photons in each bin of the original data cube.

Comparisons between the original and reconstructed hyperspectral data cube are also shown in Table 5.1. Each row contains data for the wavelength bin with a center wavelength indicated by the first column. The table shows the sum of photons in both the reconstructed and original data

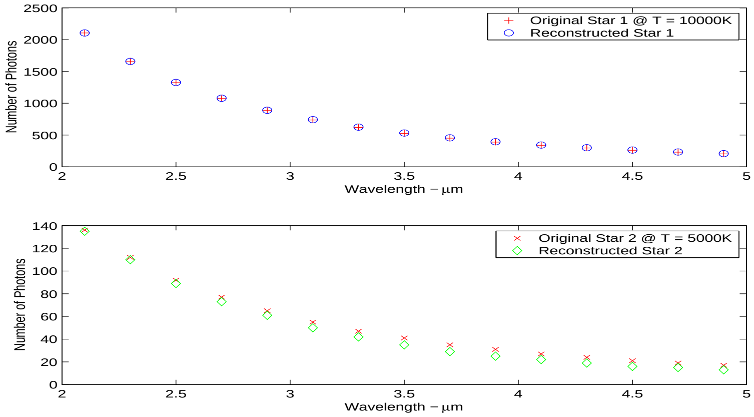


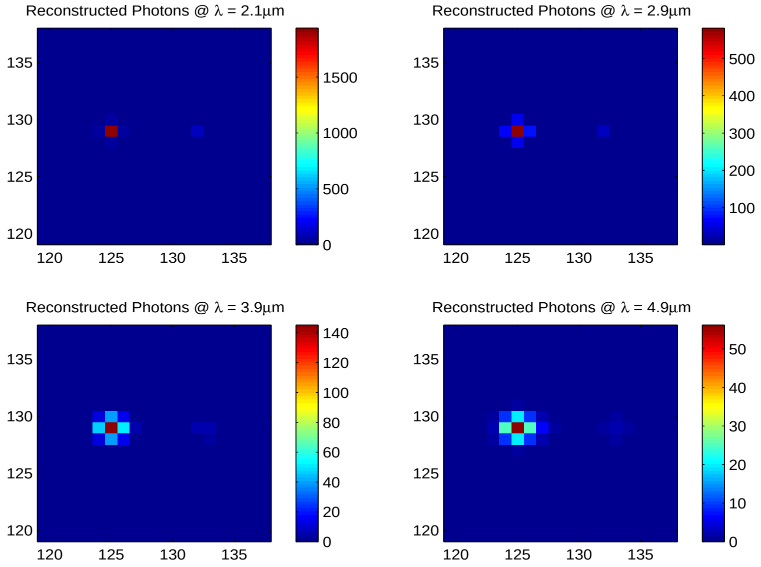
Figure 5.2: This figure shows the sum of photons in each wavelength bin for each star in the reconstructed 2D data cube compared with the original data cube.

cubes for each star. The percentage difference of the reconstruction sum compared to the original sum is also listed. The REM, labelled Error, is shown in units of *photons* and as a percentage comparison to the number of photons in the original data cube.

Table 5.1: Binary Star 2D Reconstruction Results

Bin λ	**Recon** **Star 1**	**Orig** **Star 1**	**R/O** **%**	**Recon** **Star 2**	**Orig** **Star 2**	**R/O** **%**	**Error** **Photons**	**Error** **%**
2.1	2105	2105	100.0	135	136	99.3	369	16.5
2.3	1658	1655	100.2	110	112	98.2	617	34.9
2.5	1327	1324	100.2	89	92	96.7	663	46.8
2.7	1079	1075	100.4	73	77	94.8	697	60.5
2.9	889	885	100.5	61	65	93.8	729	76.7
3.1	742	737	100.7	50	55	90.9	715	90.3
3.3	625	620	100.8	42	47	89.4	708	106.1
3.5	532	527	100.9	35	41	85.4	657	115.7
3.7	457	451	101.3	29	35	82.9	599	123.3
3.9	395	389	101.5	25	31	80.6	542	129.0
4.1	343	338	101.5	22	27	81.5	487	133.4
4.3	301	296	101.7	19	24	79.2	443	138.4
4.5	265	260	101.9	16	21	76.2	402	143.1
4.7	234	230	101.7	15	19	78.9	369	148.2
4.9	208	204	102.0	13	17	76.5	325	147.1

The results for test 4.1.2 are obtained by performing 100 iterations of the 2D atmospheric reconstruction algorithm. Four wavelength bins of the reconstructed input hyperspectral data cube, $\widehat{O}(\lambda, u, v)$, are shown in Figure 5.3. With the inclusion of the atmosphere the reconstructed data cube still exhibits similar spatial features to the original data cube. Once again additional blurring is induced by the CTIS optics and is more pronounced at the higher wavelengths.


Reconstructed Photons @ λ = 2.1μm

Reconstructed Photons @ λ = 2.9μm

Reconstructed Photons @ λ = 3.9μm

Reconstructed Photons @ λ = 4.9μm

Figure 5.3: This figure shows four wavelength bins of the reconstructed input hyperspectral data cube obtained by performing 100 iterations of the 2D atmospheric reconstruction algorithm.

The sum of photons in each wavelength bin for each star is shown in Figure 5.4 indicating that 2D atmospheric reconstruction algorithm is also, for the most part, tracking the total number of photons in each bin of the original data cube. Two bins with a low atmospheric transmission coefficient, $t_{atm}(2.7\mu m) = 0.0308$ and $t_{atm}(2.7\mu m) = 0.0003$, show degraded reconstruction results which is more evident in the lower photon numbers of Star 2. However the $\lambda = 2.7\mu m$ bin, with only a 3% atmospheric transmission, does reconstruct for Star 1. Also, as indicated by Table 5.2, the REM is actually slightly lower for each wavelength bin than for the 2D reconstruction conducted on the pristine data cube shown in Table 5.1.

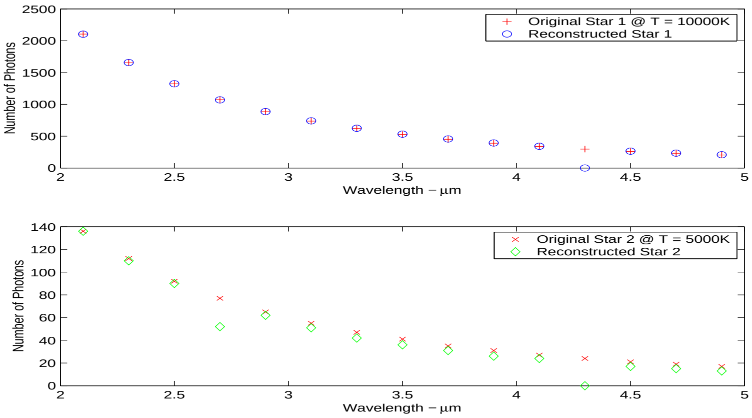


Figure 5.4: This figure shows the sum of photons in each wavelength bin for each star in the reconstructed 2D data cube compared with the original data cube when atmospheric attenuation is included.

Table 5.2: Binary Star 2D Atmospheric Reconstruction Results

Bin λ	**Recon** **Star 1**	**Orig** **Star 1**	**R/O** **%**	**Recon** **Star 2**	**Orig** **Star 2**	**R/O** **%**	**Error** **Photons**	**Error** **%**
2.1	2104	2105	100.0	136	136	100.0	366	16.3
2.3	1658	1655	100.2	110	112	98.2	590	33.4
2.5	1325	1324	100.1	90	92	97.8	600	42.4
2.7	1070	1075	99.5	52	77	67.5	854	74.1
2.9	887	885	100.2	62	65	95.4	664	69.9
3.1	741	737	100.5	51	55	92.7	690	87.1
3.3	625	620	100.8	42	47	89.4	697	104.5
3.5	532	527	100.9	36	41	87.8	648	114.1
3.7	456	451	101.1	31	35	88.6	587	120.8
3.9	393	389	101.0	26	31	83.9	531	126.4
4.1	342	338	101.2	24	27	88.9	456	124.9
4.3	0	296	0.0	0	24	0.0	320	100.0
4.5	265	260	101.9	17	21	81.0	378	134.5
4.7	234	230	101.7	15	19	78.9	360	144.6
4.9	208	204	102.0	13	17	76.5	322	145.7

The results for test 4.1.3 are obtained by performing 1000 iterations of the vector reconstruction algorithm. The ability to effectively conduct more iterations with the vector reconstruction algorithm is attributed to its faster processing time over the 2D reconstruction algorithm. As an indication, the vector algorithm processes each iteration in approximately $0.2sec$ while the 2D algorithm requires about $48sec$ per iteration. This equates to the vector algorithm being approximately 240 times faster than the 2D algorithm. The vectors in four wavelength bins of the original and reconstructed input hyperspectral data cube, $\widehat{O}(\lambda, u)$, are shown in Figure 5.5.

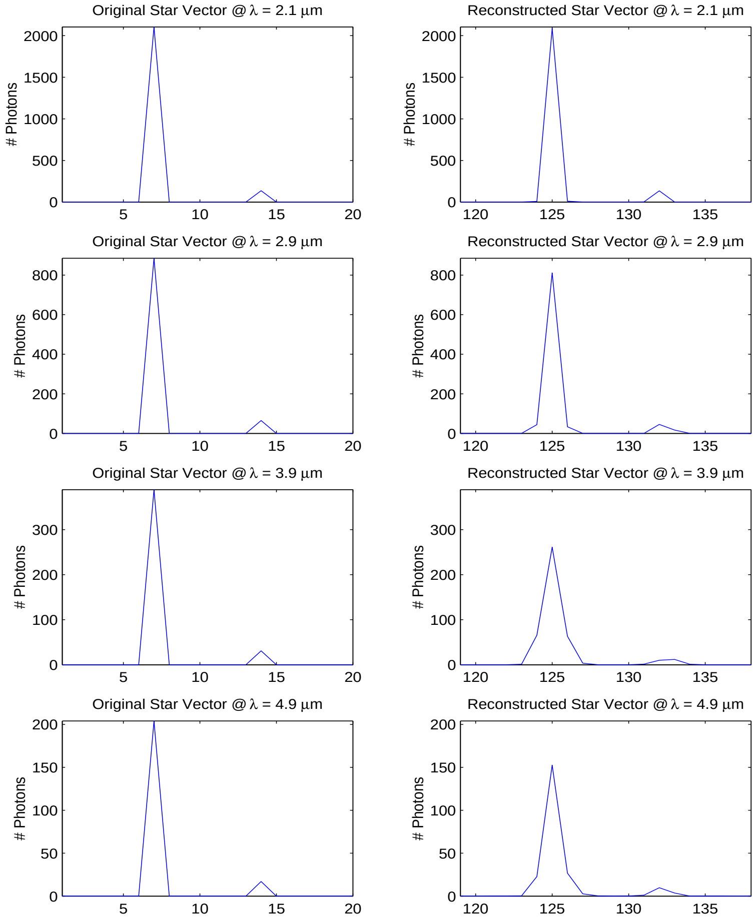


Figure 5.5: This figure shows the original and reconstructed x vector containing a horizontal sum of the number of photons in the input hyperspectral data cube. The results are obtained using the 1000 iterations of the vector reconstruction algorithm.

The sum of photons in each wavelength bin and for each star is shown in Figure 5.6 with the data also presented in Table 5.3.

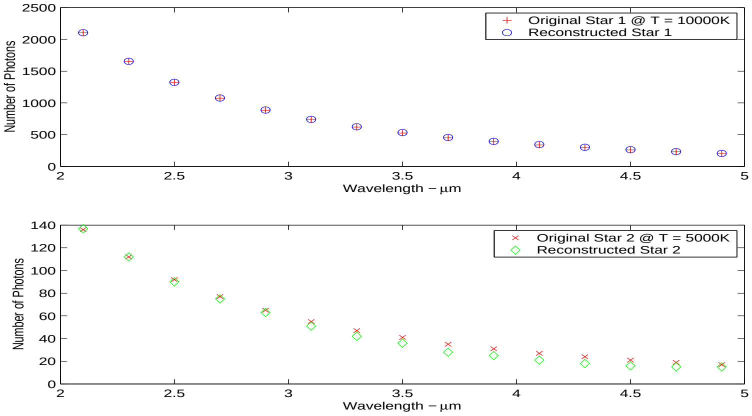


Figure 5.6: This figure shows the sum of photons in each wavelength bin for each star in the reconstructed 1D data matrix compared with the original data matrix.

Table 5.3: Binary Star Vector Reconstruction Results

Bin λ	**Recon Star 1**	**Orig Star 1**	**R/O %**	**Recon Star 2**	**Orig Star 2**	**R/O %**	**Error Photons**	**Error %**
2.1	2105	2105	100.0	137	136	100.7	41	1.8
2.3	1655	1655	100.0	112	112	100.0	75	4.2
2.5	1325	1324	100.1	90	92	97.8	106	7.5
2.7	1077	1075	100.2	75	77	97.4	142	12.3
2.9	888	885	100.3	63	65	96.9	188	19.8
3.1	741	737	100.5	51	55	92.7	257	32.4
3.3	625	620	100.8	42	47	89.4	287	43.0
3.5	533	527	101.1	36	41	87.8	294	51.8
3.7	457	451	101.3	28	35	80.0	314	64.6
3.9	395	389	101.5	25	31	80.6	297	70.7
4.1	344	338	101.8	21	27	77.8	284	77.8
4.3	302	296	102.0	18	24	75.0	272	85.0
4.5	265	260	101.9	16	21	76.2	242	86.1
4.7	234	230	101.7	15	19	78.9	219	88.0
4.9	206	204	101.0	15	17	88.2	117	52.9

The results for test 4.1.4 are obtained by performing 1000 iterations of the vector atmospheric reconstruction algorithm. The sum of photons in each wavelength bin and for each star is shown in Figure 5.7 with the data also presented in Table 5.4. As with the 2D atmospheric case, the reconstruction is degraded for the bins with low atmospheric transmission coefficients. However the remaining bins provide a similar reconstruction performance to the 2D atmospheric case.

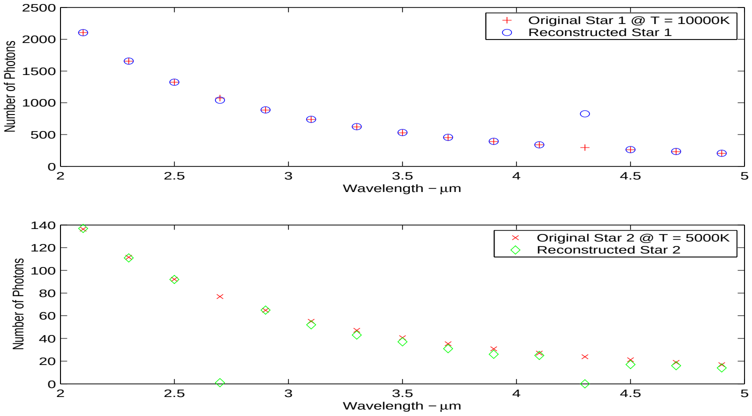


Figure 5.7: This figure shows the original and reconstructed x vector containing a horizontal sum of the number of photons in the input hyperspectral data cube. The results are obtained using 1000 iterations of the vector atmospheric reconstruction algorithm.

Table 5.4: Binary Star Vector Atmospheric Reconstruction Results

Bin λ	**Recon Star 1**	**Orig Star 1**	**R/O %**	**Recon Star 2**	**Orig Star 2**	**R/O %**	**Error Photons**	**Error %**
2.1	2104	2105	100.0	137	136	100.5	44	2.0
2.3	1656	1655	100.1	111	112	99.5	97	5.5
2.5	1325	1324	100.1	92	92	99.8	133	9.4
2.7	1041	1075	96.8	1	77	1.3	411	35.7
2.9	888	885	100.3	65	65	100.3	152	16.0
3.1	740	737	100.4	52	55	94.5	212	26.8
3.3	625	620	100.8	43	47	91.3	272	40.8
3.5	532	527	101.0	37	41	91.1	264	46.5
3.7	456	451	101.1	31	35	87.9	277	57.0
3.9	393	389	101.0	26	31	84.4	247	58.8
4.1	339	338	100.3	25	27	92.5	158	43.3
4.3	827	296	279.8	0	24	0.0	1145	358.5
4.5	263	260	101.2	17	21	80.3	185	65.8
4.7	234	230	101.9	16	19	84.9	202	81.3
4.9	206	204	101.0	14	17	83.0	114	51.6

A summary of the results for the binary star test cases 4.1.1 to 4.1.4 is presented in Table 5.5. The results indicate no degradation in reconstruction performance when the vector algorithm is used over the 2D reconstruction algorithm. The results for the atmospheric reconstruction algorithm on atmospherically attenuated detector data also maintain accuracy when compared to the pristine detector data. The table indicates the atmospheric reconstruction results when the low atmospheric transmission coefficient bins are not considered. The corresponding values for these bins are also shown.

Table 5.5: Binary Star Results Summary

	Photon Sum Error %		**REM**
Test Case	**Star 1**	**Star 2**	**%**
2D	0.0 - 2.0	0.7 - 23.5	16.5 - 148.2
2D + atm + mask	0.0 - 2.0	0.0 - 23.5	16.3 - 145.7
2D + atm: bad bins	100, 0.5	32.5, 100	69.9, 100
1D	0.0 - 2.0	0.0 - 25.0	1.8 - 88.0
1D + atm + mask	0.0 - 1.9	0.3 - 19.7	2.0 - 81.3
1D + atm: bad bins	3.2, 179.8	98.7, 100.0	35.7, 358.5

The temperatures of each star are also estimated from the reconstructed photon data for each test case as shown in Table 5.6. The table shows the estimate obtained by considering only the spectral data at the peak pixel spatial location for each star. Due to the blurring induced by the modelled optics this temperature can be significantly lower than the respective original star temperature. Improvement in the estimation is achieved by summing the received photons for each star prior to performing the temperature estimation. The validity of using this form of post-processing is applicable to the binary star scenario but also assumes some prior knowledge of the distribution of photons arriving from the individual stars. The table also shows the temperature estimation improvement when removing, or masking out, the bins with low atmospheric transmission coefficients.

Table 5.6: Binary Star Reconstructed Temperatures

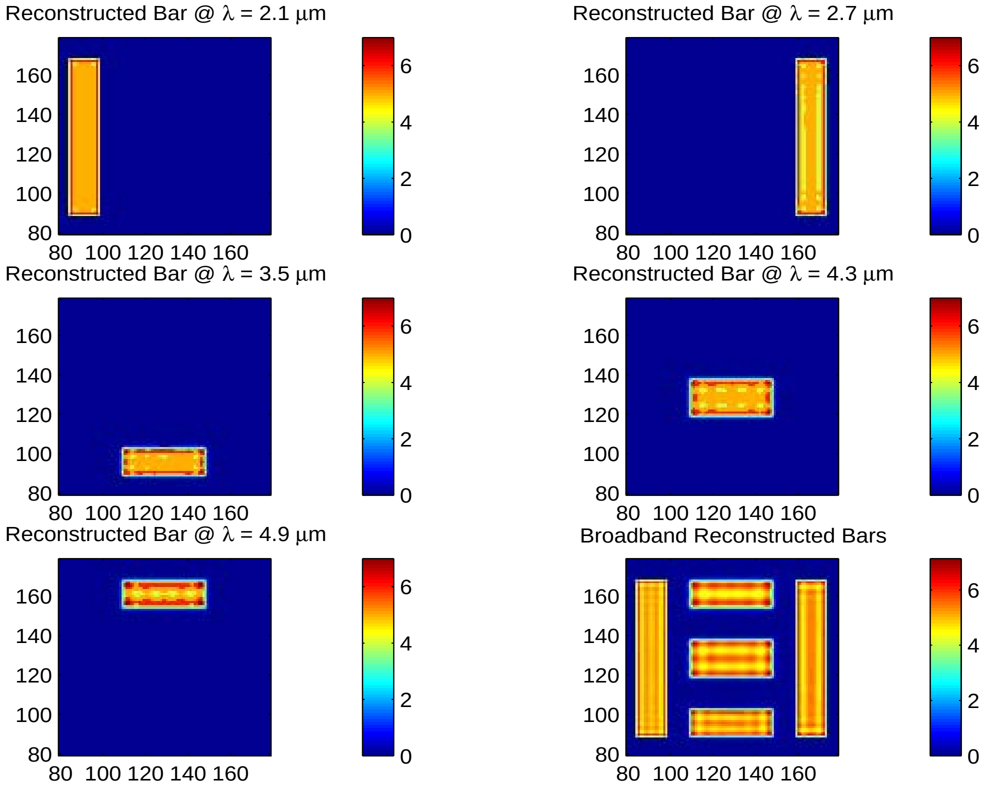
	Reconstructed Star 1 Temp. (Original at 10000°K)				**Reconstructed Star 2 Temp. (Original at 5000°K)**			
Test Case	**Peak pixel °K**	**Error %**	**Summed pixel °K**	**Error %**	**Peak pixel °K**	**Error %**	**Summed pixel °K**	**Error %**
2D	8324	-16.8	10022	0.2	3948	-21.0	4873	-2.5
2D + atm	8322	-16.8	9964	-0.4	3953	-20.9	4804	-3.9
2D + atm + mask	8473	-15.3	10015	0.2	4095	-18.1	4909	-1.8
1D	9590	-4.1	10015	0.2	4527	-9.5	4920	-1.6
1D + atm	9499	-5.0	10070	0.7	4369	-12.6	4666	-6.7
1D + atm + mask	9624	-3.8	10010	0.1	4942	-1.2	4959	-0.8

5.3 Spatially Separate Monochromatic Source Results

The results for tests 4.2.1 and 4.2.2 are obtained by separately performing 100 iterations of the 2D reconstruction algorithm for each test case. The clean image at the detector is used in test 4.2.1 to reconstruct the input data cube with the results shown in Figure 5.8.

Test 4.2.2 adds noise to the detector image prior to performing the reconstruction. Though the reconstructed data is degraded as shown in Figure 5.9, the shape of the original source bars is still apparent. The degradation of the reconstruction is quantified in Figure 5.10 which shows the sum of photons in each wavelength bin for the original, noiseless reconstructed and noisy reconstructed data.

This effect of detector noise on the 2D reconstruction is also presented in Table 5.7 which shows the sum of photons in each reconstructed bin for both the clean and noisy detector data. This is also displayed as a percentage compared to the number of photons in the bins of the original data cube. The REM is also shown in units of *photons* and as a percentage for the bins containing sources. It is observed that the REM is significantly lower for the extended sources in these test cases compared to the point sources in the binary star test case results in section 5.2.


Reconstructed Bar @ λ = 2.1 µm

Reconstructed Bar @ λ = 2.7 µm

Reconstructed Bar @ λ = 3.5 µm

Reconstructed Bar @ λ = 4.3 µm

Reconstructed Bar @ λ = 4.9 µm

Broadband Reconstructed Bars

Figure 5.8: This figure shows the reproduction of the original bar test data cube by the 2D reconstruction algorithm. The bar test consists of five monochromatic sources in five spectral bins. The original sources have a uniform five photons across their extent with the remaining bins containing zero photons. The final subplot shows a total $2 - 5\mu m$ view of the reproduced data.

Table 5.7: Spatially Separated Monochromatic Source 2D Reconstruction Results for Clean and Noisy Data

Bin **λ**	**Orig** **Bar**	**Recon** **Bar**	**Recon** **Bar N**	**R/O** **%**	**R/O** **% N**	**Error** **Photons**	**Error** **Photons N**	**Error** **%**	**Error** **% N**
2.1	6000	5932	5867	98.9	97.8	378	1438	6.3	24.0
2.3	0	69	118			0	49		
2.5	0	101	129			0	50		
2.7	6000	5794	5738	96.6	95.6	754	1292	12.6	21.5
2.9	0	106	132			0	50		
3.1	0	0	0			0	0		
3.3	0	31	36			0	9		
3.5	3000	2944	2938	98.1	97.9	453	695	15.1	23.2
3.7	0	24	38			0	10		
3.9	0	0	0			0	0		
4.1	0	23	38			0	6		
4.3	4000	3956	3942	98.9	98.6	541	763	13.5	19.1
4.5	0	18	18			0	0		
4.7	0	1	2			0	0		
4.9	3000	3001	2989	100.0	99.6	583	719	19.4	24.0

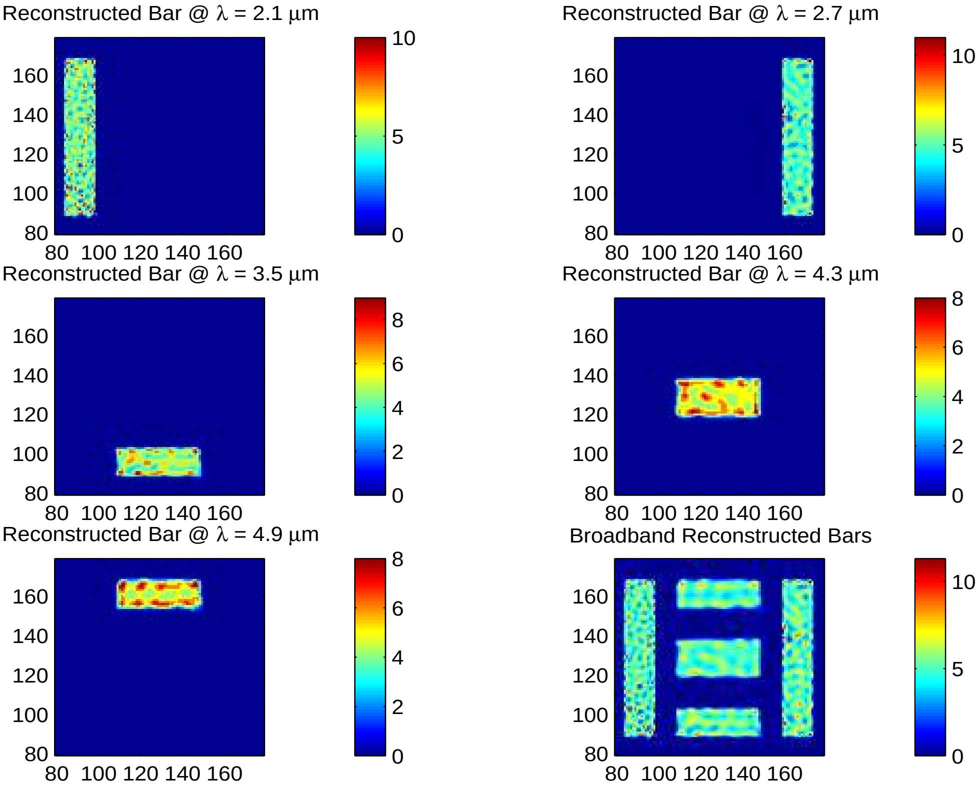


Figure 5.9: This figure shows the degraded performance of the 2D reconstruction algorithm in the presence of noisy detector data. The spatial features of the original data remain in the reconstruction of the noisy data.

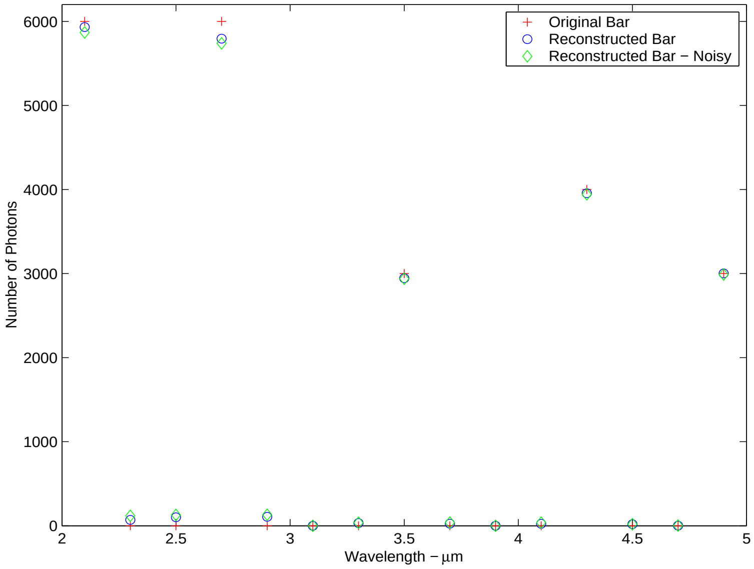


Figure 5.10: This figure shows the sum of photons in each wavelength bin for the bar test in the reconstructed 2D data cube compared with the original data cube. The sum of photons for the reconstruction of noisy detector data is also shown.

The results for tests 4.2.3 and 4.2.4 are obtained by separately performing 1000 iterations of the vector reconstruction algorithm for each test case. The clean image at the detector is used in test 4.2.3 while a noisy detector image is used in test 4.2.4. The sources in the reconstructed input data cube for both test cases are shown in Figure 5.11.

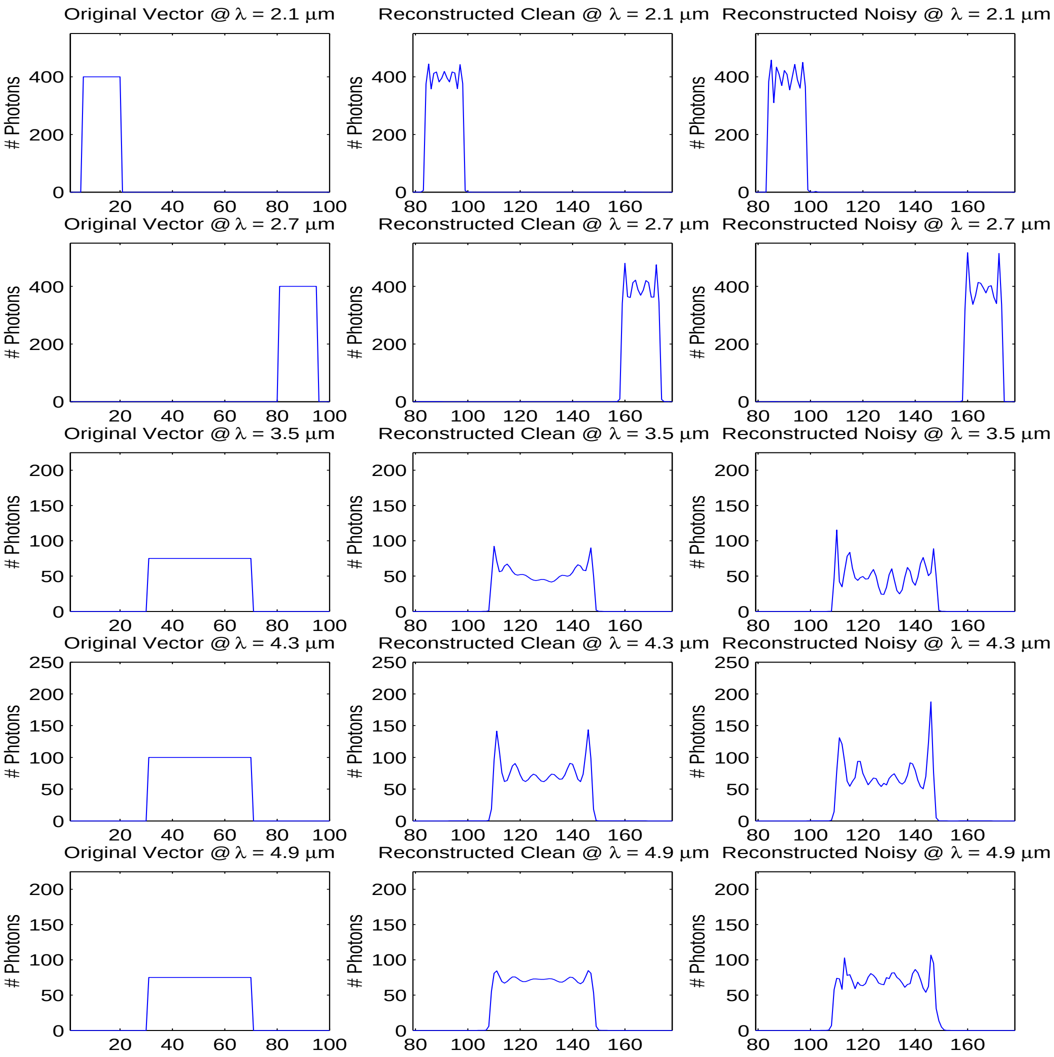


Figure 5.11: This figure shows the reproduction of the original bar test data cube by the 1D reconstruction algorithm. The reconstruction of x vector sum degrades with the addition of noise to the detector image, though the general spatial features are still apparent.

The photon sums in each bin are shown in Figure 5.12. The difference in performance between the 2D and vector reconstruction algorithms is also observed by comparing Tables 5.8 and 5.7.

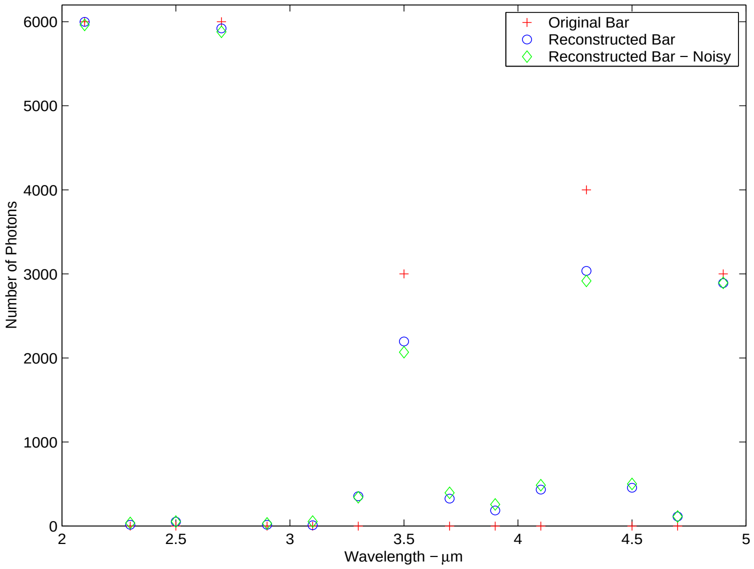


Figure 5.12: This figure shows the sum of photons in each wavelength bin for the bar test in the reconstructed data compared with the original data cube. The sum of photons for the reconstruction of noisy detector data is also shown. The breakdown of the reconstruction is evident in the higher wavelength bins and increased photon numbers in the originally empty bins.

Table 5.8: Spatially Separated Monochromatic Source Vector Reconstruction Results for Clean and Noisy Data

Bin λ	**Orig** **Bar**	**Recon** **Bar**	**Recon** **Bar N**	**R/O** **%**	**R/O** **% N**	**Error** **Photons**	**Error** **Photons N**	**Error** **%**	**Error** **% N**
2.1	6000	5998	5962	100.0	99.4	352	503	5.9	8.4
2.3	0	16	38			17	36		
2.5	0	52	54			52	55		
2.7	6000	5921	5880	98.7	98.0	559	645	9.3	10.8
2.9	0	15	31			15	33		
3.1	0	9	53			7	53		
3.3	0	354	342			353	339		
3.5	3000	2196	2068	73.2	68.9	877	1067	29.2	35.6
3.7	0	326	396			322	394		
3.9	0	185	258			185	257		
4.1	0	434	487			435	485		
4.3	4000	3036	2917	75.9	72.9	1170	1406	29.3	35.2
4.5	0	456	501			455	502		
4.7	0	113	113			111	112		
4.9	3000	2890	2893	96.3	96.4	208	426	6.9	14.2

The results for tests 4.2.5 and 4.2.6 are obtained by separately performing 1000 iterations of the vector reconstruction algorithm for each test case. The major distinction between this and the previous test case is that the smaller spatial extent (20 × 20 pixels) of each wavelength source in the input data is completely spatially dispersed at the detector. The clean image at the detector is used in test 4.2.5 while a noisy detector image is used in test 4.2.6. The sources in the reconstructed input data cube for both test cases are shown in Figure 5.13.

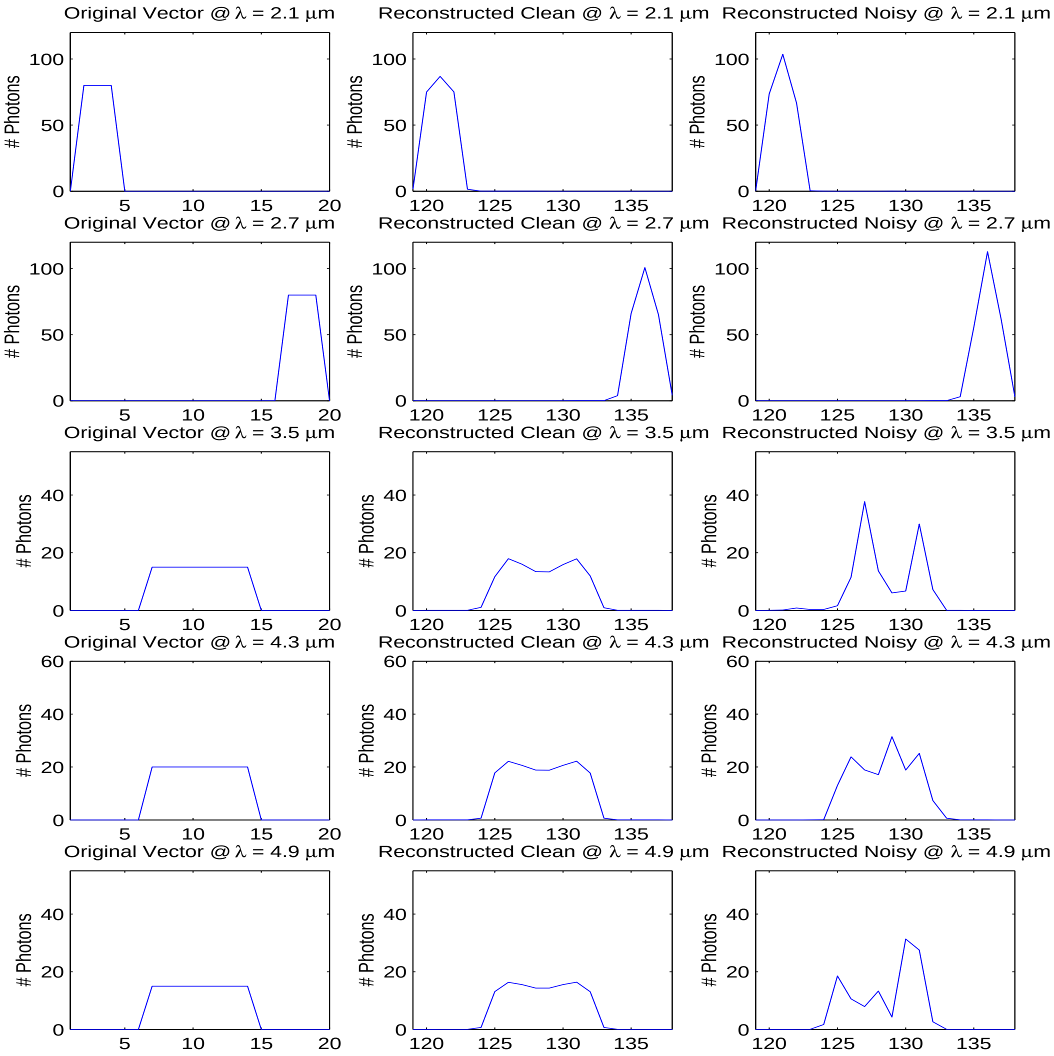


Figure 5.13: This figure shows the reproduction of the original small bar test data cube by the 1D reconstruction algorithm. The reconstruction of x vector sum suffers more degradation than the previous test case with the addition of noise to the detector image.

The photon sums in each bin are shown in Figure 5.14 indicate improved performance over the previous test case. However the remaining reconstruction results in Table 5.9, the reconstruction results show mixed REM results in comparison to Table 5.8, but generally confirm a high susceptibility to noise with the smaller input data cube.

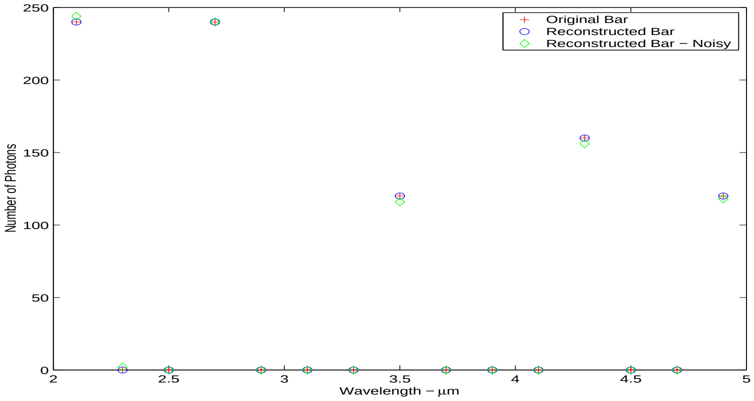


Figure 5.14: This figure shows the sum of photons in each wavelength bin for the smaller bar test in the reconstructed data compared with the original data cube. The sum of photons for the reconstruction of noisy detector data is also shown. The smaller input data cube leads to improved reconstruction performance in terms of reproducing the number of photons in the original scene.

Table 5.9: Spatially Separated Monochromatic Source Vector Reconstruction Results For Clean And Noisy Data Using Small Input Data Cube

Bin	**Orig**	**Recon**	**Recon**	**R/O**	**R/O**	**Error**	**Error**	**Error**	**Error**
λ	**Bar**	**Bar**	**Bar N**	**%**	**% N**	**Photons**	**Photons N**	**%**	**% N**
2.1	240	240	244	100.0	101.7	20	43	8.3	17.9
2.3	0	0	2			0	1		
2.5	0	0	0			0	0		
2.7	240	240	240	100.0	100.0	58	82	24.2	34.2
2.9	0	0	0			0	0		
3.1	0	0	0			0	0		
3.3	0	0	0			0	0		
3.5	120	120	116	100.0	96.7	20	82	16.7	68.3
3.7	0	0	0			0	0		
3.9	0	0	0			0	0		
4.1	0	0	0			0	0		
4.3	160	160	156	100.0	97.5	14	46	8.8	28.8
4.5	0	0	0			0	0		
4.7	0	0	0			0	0		
4.9	120	120	118	100.0	98.3	12	71	10.0	59.2

A summary of the results for the spatially separated monochromatic source test cases 4.2.1 to 4.2.6 is presented in Table 5.10. The table indicates little degradation in 2D and vector reconstruction performance with the addition of noise to the detector image. However a significant performance degradation is demonstrated for the vector reconstruction algorithm over each reconstruction metric. This is attributed to the increased spatial information in the scene not being fully reproduced in the corresponding vector representation. Performance improves with the smaller imaged scene, however this scene is also more susceptible to noise.

Table 5.10: Spatially Separated Monochromatic Source Results Summary

Test Case	**Photon Sum Error** %	**REM** %	**Spectral Bleeding** %
2D	0.0 - 3.4	6.3 - 19.4	0.0 - 1.8
2D + noise	0.4 - 4.4	19.1 - 24.0	0.0 - 2.2
1D (100 × 100)	0.0 - 26.8	5.9 - 29.3	0.3 - 11.8
1D + noise	0.6 - 31.1	8.4 - 35.6	0.5 - 13.2
1D (20 × 20)	0.0	8.3 - 24.2	0.0
1D + noise	0.0 - 3.3	17.9 - 68.3	0.0 - 0.8

The results for tests 4.3.1 and 4.3.2 are obtained by separately performing 100 iterations of the 2D reconstruction algorithm for each test case. The clean image at the detector is used in test 4.3.1 to reconstruct the input data cube with the results shown in Figure 5.15.

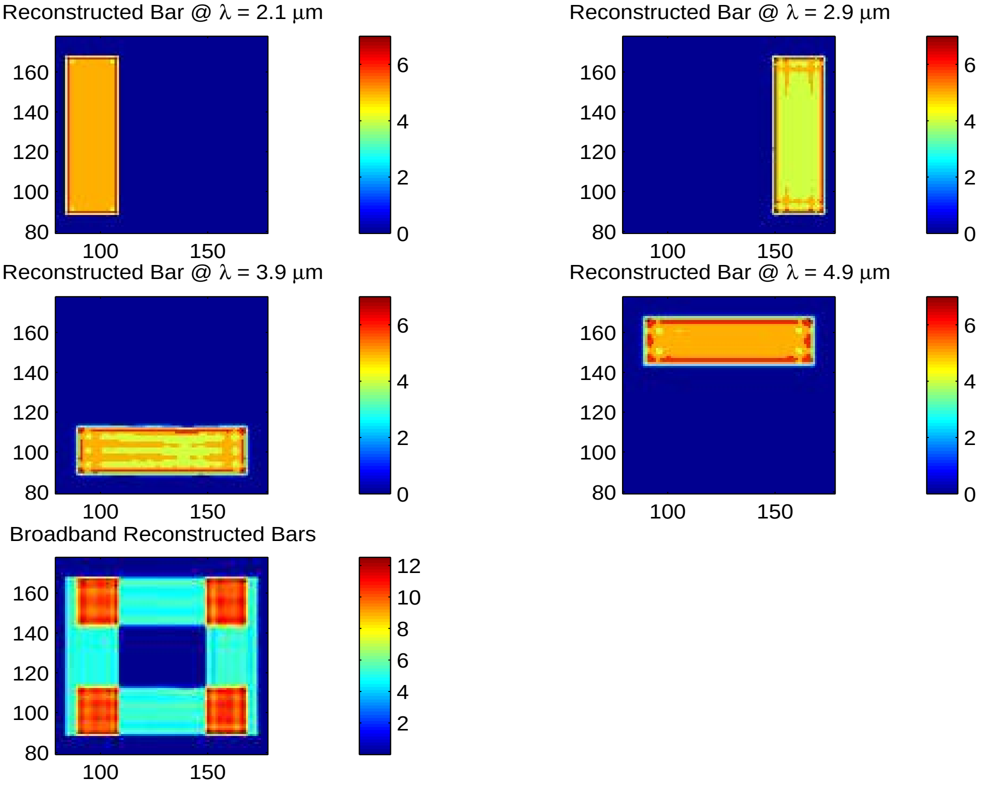


Figure 5.15: This figure shows the reproduction of the original overlapping bar test data cube by the 2D reconstruction algorithm. The bar test consists of four monochromatic sources in four spectral bins. The original sources have a uniform five photons across their extent with the remaining bins containing zero photons. The final subplot shows a total $2-5\mu m$ view of the reproduced data.

Test 4.3.2 adds noise to the detector image prior to performing the reconstruction. Though the reconstructed data is degraded as shown in Figure 5.16, the shape of the original source bars is still evident. The degradation of the reconstruction is also shown in Figure 5.17 which shows the sum of photons in each wavelength bin for the original, reconstructed and noisy reconstructed data and Table 5.11 which shows the REM.

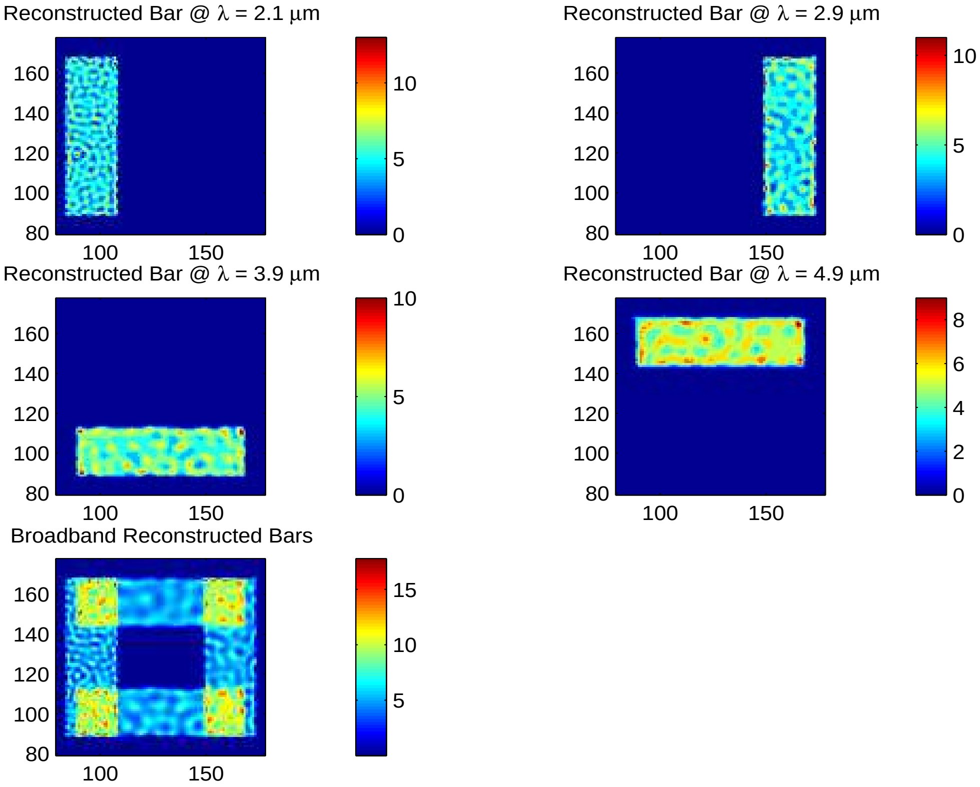


Figure 5.16: This figure shows the degraded performance of the 2D reconstruction algorithm in the presence of noisy detector data for the overlapping bar test. However the spatial features of the original data do remain in the reconstruction of the noisy data.

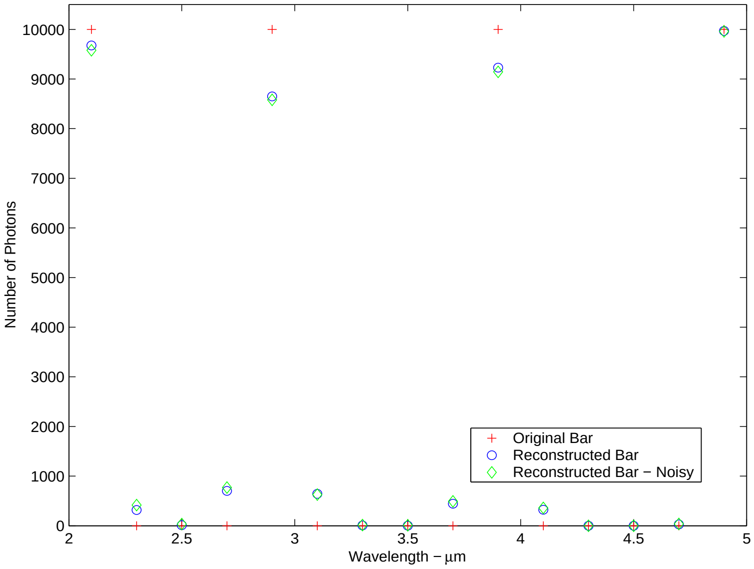


Figure 5.17: This figure shows the sum of photons in each wavelength bin for the overlapping bar test in the reconstructed 2D data cube compared with the original data cube. The sum of photons for the reconstruction of noisy detector data is also shown.

This test demonstrates the ability of the 2D reconstruction algorithm to spectrally separately spatially overlapping sources.

Table 5.11: Spatially Overlapping Monochromatic Source 2D Reconstruction Results for Clean and Noisy Data

Bin	**Orig**	**Recon**	**Recon**	**R/O**	**R/O**	**Error**	**Error**	**Error**	**Error**
λ	**Bar**	**Bar**	**Bar N**	**%**	**% N**	**Photons**	**Photons N**	**%**	**% N**
2.1	10000	9677	9580	96.8	95.8	419	2360	4.2	23.6
2.3	0	318	416			0	240		
2.5	0	10	34			0	3		
2.7	0	703	768			782	698		
2.9	10000	8651	8579	86.5	85.8	1865	2522	18.7	25.2
3.1	0	642	628			668	516		
3.3	0	7	13			0	0		
3.5	0	2	8			0	0		
3.7	0	444	492			205	353		
3.9	10000	9228	9144	92.3	91.4	1489	1927	14.9	19.3
4.1	0	321	360			42	189		
4.3	0	0	0			0	0		
4.5	0	0	0			0	0		
4.7	0	25	35			0	0		
4.9	10000	9970	9960	99.7	99.6	1029	1662	10.3	16.6

The results for tests 4.3.3 and 4.3.4 are obtained by separately performing 1000 iterations of the vector reconstruction algorithm for each test case. The clean image at the detector is used in test 4.3.3 while a noisy detector image is used in test 4.3.4. The sources in the reconstructed input data cube for both test cases are shown in Figure 5.18.

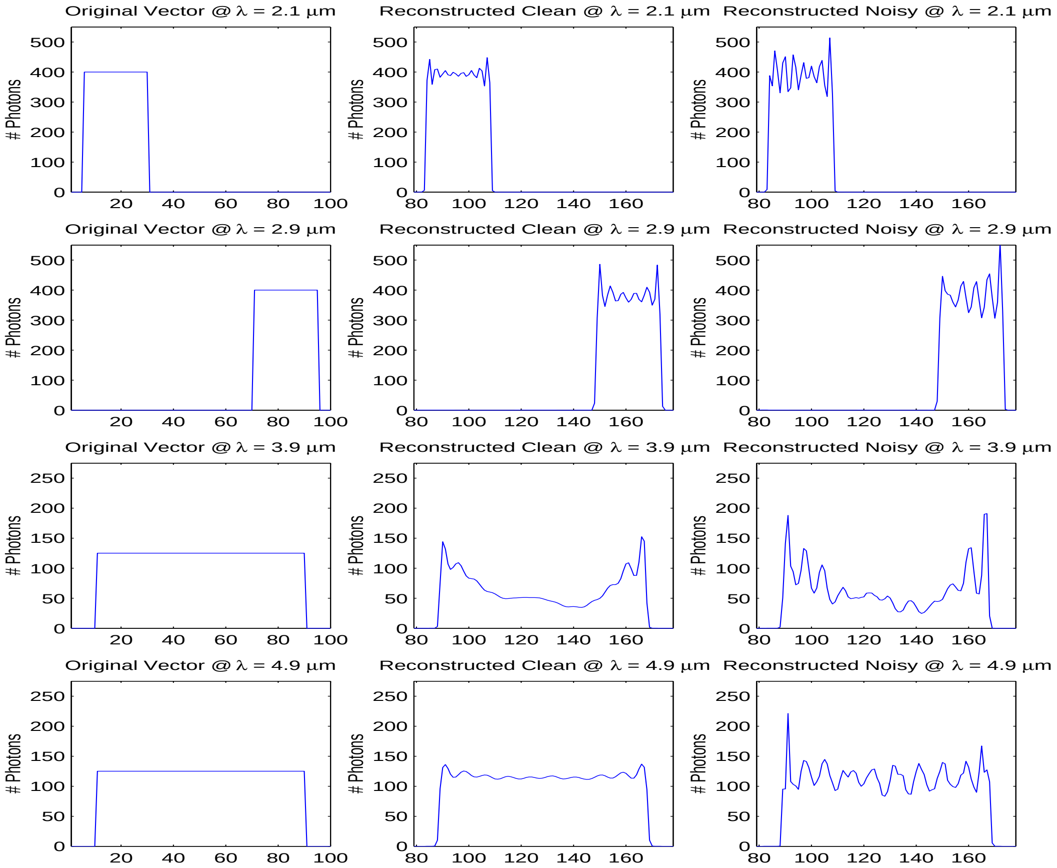


Figure 5.18: This figure shows the reproduction of the original overlapping bar test data cube by the 1D reconstruction algorithm. Again the reconstruction of x vector sum degrades with the addition of noise to the detector image.

The photon sums in each bin are shown in Figure 5.19. The difference in performance between the 2D and vector reconstruction algorithms is also observed by comparing Tables 5.12 and 5.11.

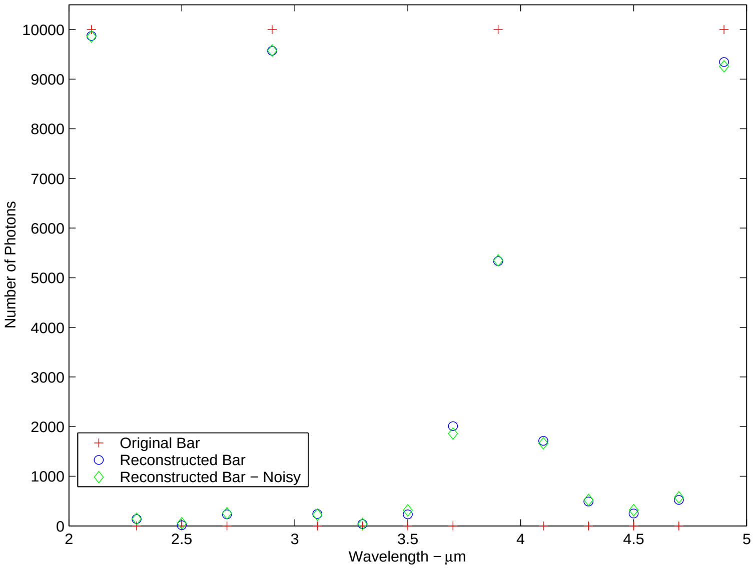


Figure 5.19: This figure shows the sum of photons in each wavelength bin for the overlapping bar test in the reconstructed data compared with the original data cube. The sum of photons for the reconstruction of noisy detector data is also shown. The breakdown of the reconstruction is evident in the higher wavelength bins and increased photon numbers in the originally empty bins.

Table 5.12: Spatially Overlapping Monochromatic Source Vector Reconstruction Results for Clean and Noisy Data

Bin λ	**Orig Bar**	**Recon Bar**	**Recon Bar N**	**R/O %**	**R/O % N**	**Error Photons**	**Error Photons N**	**Error %**	**Error % N**
2.1	10000	9870	9859	98.7	98.6	420	1072	4.2	10.7
2.3	0	138	143			136	141		
2.5	0	13	54			10	54		
2.7	0	232	255			230	255		
2.9	10000	9573	9576	95.7	95.8	883	1224	8.8	12.2
3.1	0	241	225			241	224		
3.3	0	35	39			34	37		
3.5	0	232	315			233	314		
3.7	0	2009	1861			2007	1861		
3.9	10000	5335	5350	53.4	53.5	4820	5134	48.2	51.3
4.1	0	1713	1664			1713	1659		
4.3	0	492	522			495	524		
4.5	0	249	318			248	316		
4.7	0	524	575			525	574		
4.9	10000	9345	9258	93.5	92.6	789	1389	7.9	13.9

The results for tests 4.3.5 and 4.3.6 are obtained by separately performing 1000 iterations of the vector reconstruction algorithm for each test case. Again the distinction between this and the previous test case is the smaller spatial extent (20 × 20 pixels) of each wavelength source and the complete spatial dispersion of the sources at the detector. The clean image at the detector is used in test 4.3.5 while a noisy detector image is used in test 4.3.6. The sources in the reconstructed input data cube for both test cases are shown in Figure 5.20.

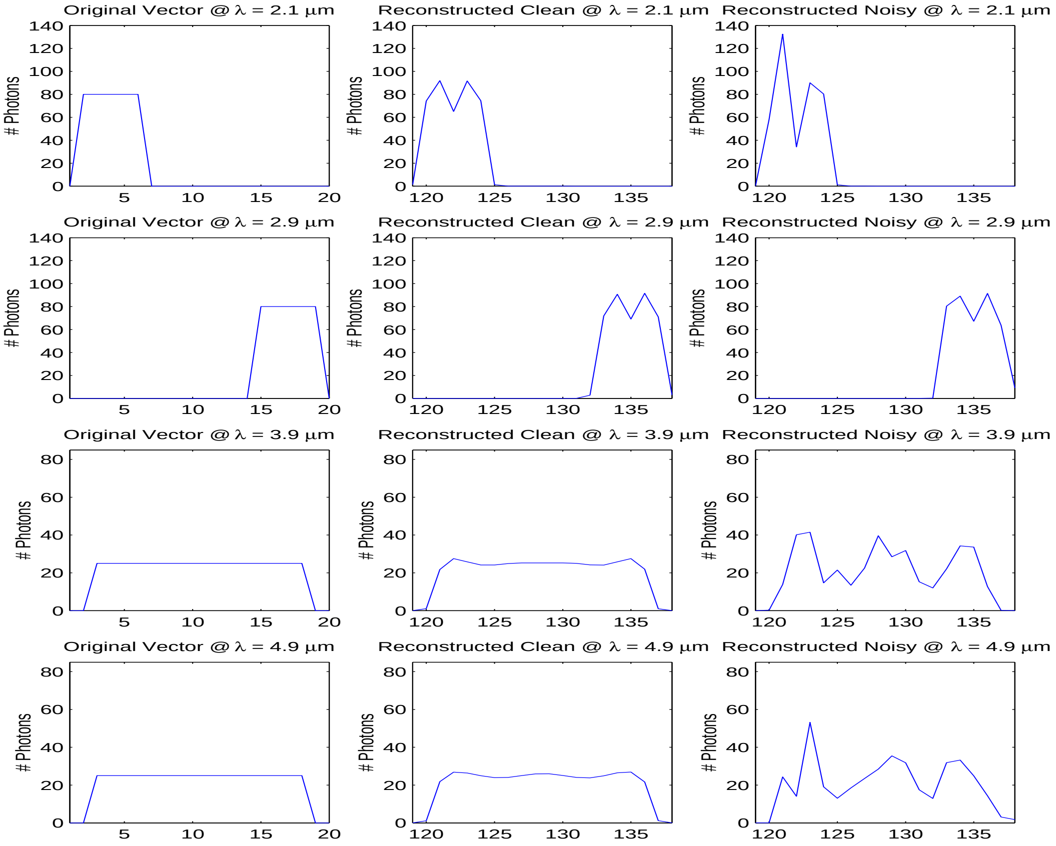


Figure 5.20: This figure shows the reproduction of the original small bar test data cube by the 1D reconstruction algorithm. The reconstruction of x vector sum suffers more degradation than the previous test case with the addition of noise to the detector image.

The photon sums in each bin are shown in Figure 5.21 indicate improved performance over the previous test case. However the remaining reconstruction results in Table 5.13, the reconstruction results show mixed REM results in comparison to Table 5.12, but generally confirm a high susceptibility to noise with the smaller input data cube.

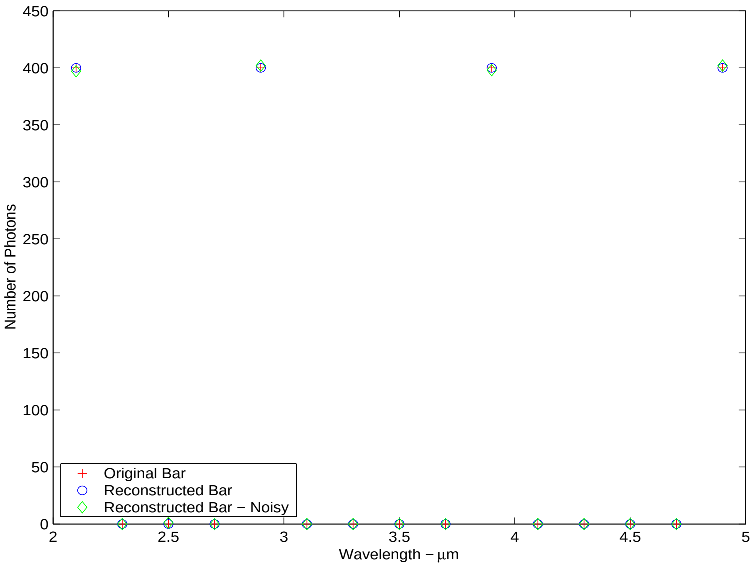


Figure 5.21: This figure shows the sum of photons in each wavelength bin for the smaller overlapping bar test in the reconstructed data compared with the original data cube. The sum of photons for the reconstruction of noisy detector data is also shown. The smaller input data cube leads to improved reconstruction performance in terms of reproducing the number of photons in the original scene.

Table 5.13: Spatially Overlapping Monochromatic Source Vector Reconstruction Results For Clean And Noisy Data Using Small Input Data Cube

Bin λ	**Orig Bar**	**Recon Bar**	**Recon Bar N**	**R/O %**	**R/O % N**	**Error Photons**	**Error Photons N**	**Error %**	**Error % N**
2.1	400	400	397	100.0	99.3	53	131	13.3	32.8
2.3	0	0	0			0	0		
2.5	0	0	1			0	0		
2.7	0	0	0			0	0		
2.9	400	400	402	100.0	100.5	57	58	14.3	14.5
3.1	0	0	0			0	0		
3.3	0	0	0			0	0		
3.5	0	0	0			0	0		
3.7	0	0	0			0	0		
3.9	400	400	398	100.0	99.5	20	152	5.0	38.0
4.1	0	0	0			0	0		
4.3	0	0	0			0	0		
4.5	0	0	0			0	0		
4.7	0	0	0			0	0		
4.9	400	400	402	100.0	100.5	20	137	5.0	34.3

A summary of the results for the spatially overlapping monochromatic source test cases 4.3.1 to 4.3.6 is presented in Table 5.14. The table again indicates little degradation in 2D and vector reconstruction performance with the addition of noise to the detector image. It is observed that Photon Sum Error and Spectral Bleeding have slightly increased in comparison to the spatially separated monochromatic source test cases. The results from the vector reconstruction again show significant performance degradation compared to the 2D reconstruction results. Performance again improves with the smaller imaged scene, but with increased noise susceptibility.

Table 5.14: Spatially Overlapping Monochromatic Source Results Summary

Test Case	**Photon Sum Error** %	**REM** %	**Spectral Bleeding** %
2D	0.3 - 13.5	4.2 - 18.7	0.0 - 6.3
2D + noise	0.4 - 18.7	16.6 - 25.2	0.0 - 6.4
1D (100 × 100)	1.3 - 46.6	4.2 - 48.2	0.1 - 20.1
1D + noise	1.4 - 46.5	10.7 - 51.3	0.5 - 18.6
1D (20 × 20)	0.0	5.0 - 14.3	0.0
1D + noise	0.5 - 0.7	14.5 - 38.0	0.0 - 0.3

5.5 *Monochromatic Numbers Source Results*

The results for test 4.4.1 are obtained by performing 100 iterations of the 2D reconstruction algorithm. Generated from a clean detector image, each wavelength bin of the reconstructed input data cube is shown in Figure 5.22. The reconstructed data cube exhibits similar spatial features to the original data cube in Figure 4.21 with some spectral overlap evident in adjacent bins.

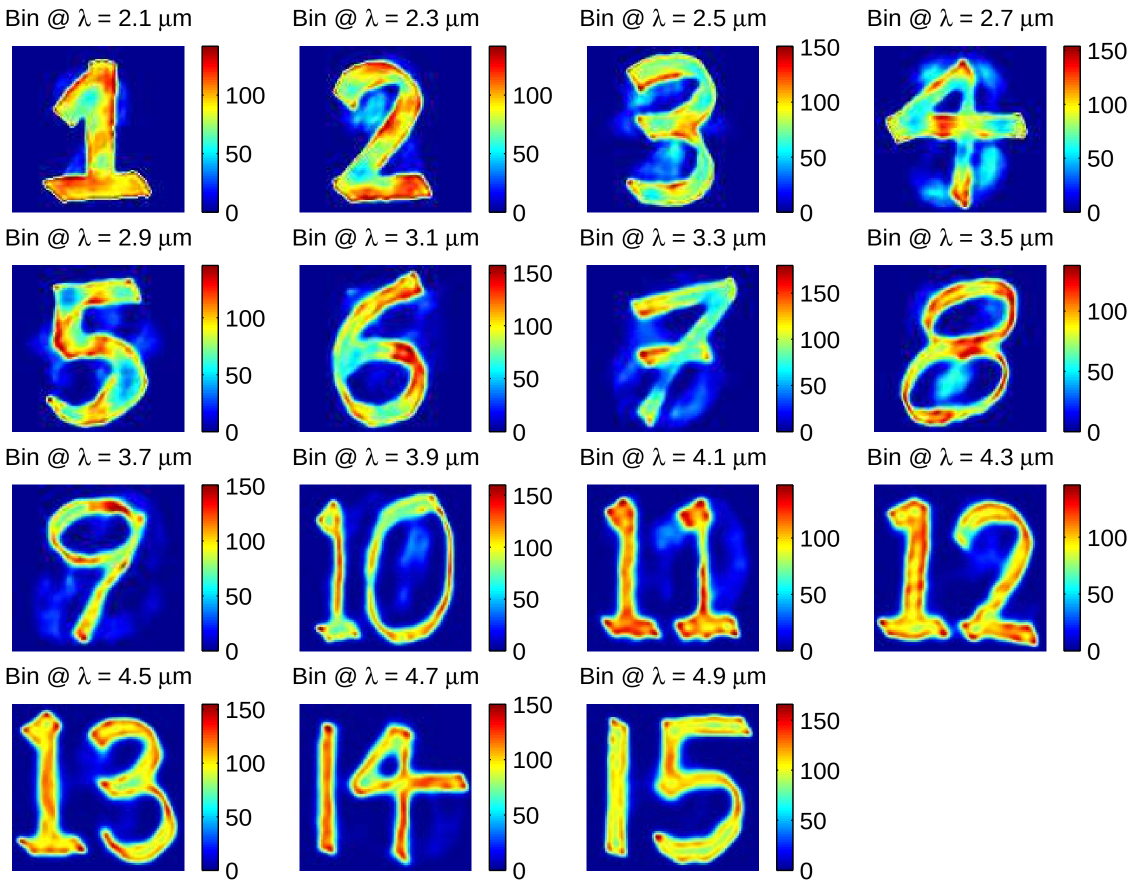


Figure 5.22: This figure shows the reconstruction of the noiseless numbers test data cube by the 2D reconstruction algorithm. The original monochromatic sources have a uniform hundred photons across their extent.

Test 4.4.2 uses a noisy detector image and reconstructs the input data cube using 100 iterations of the 2D reconstruction algorithm. The 2D reconstruction in the presence of noise is highly comparable to the noiseless reconstruction as shown in Figure 5.23 and indicated in Tables 5.15 and 5.16.

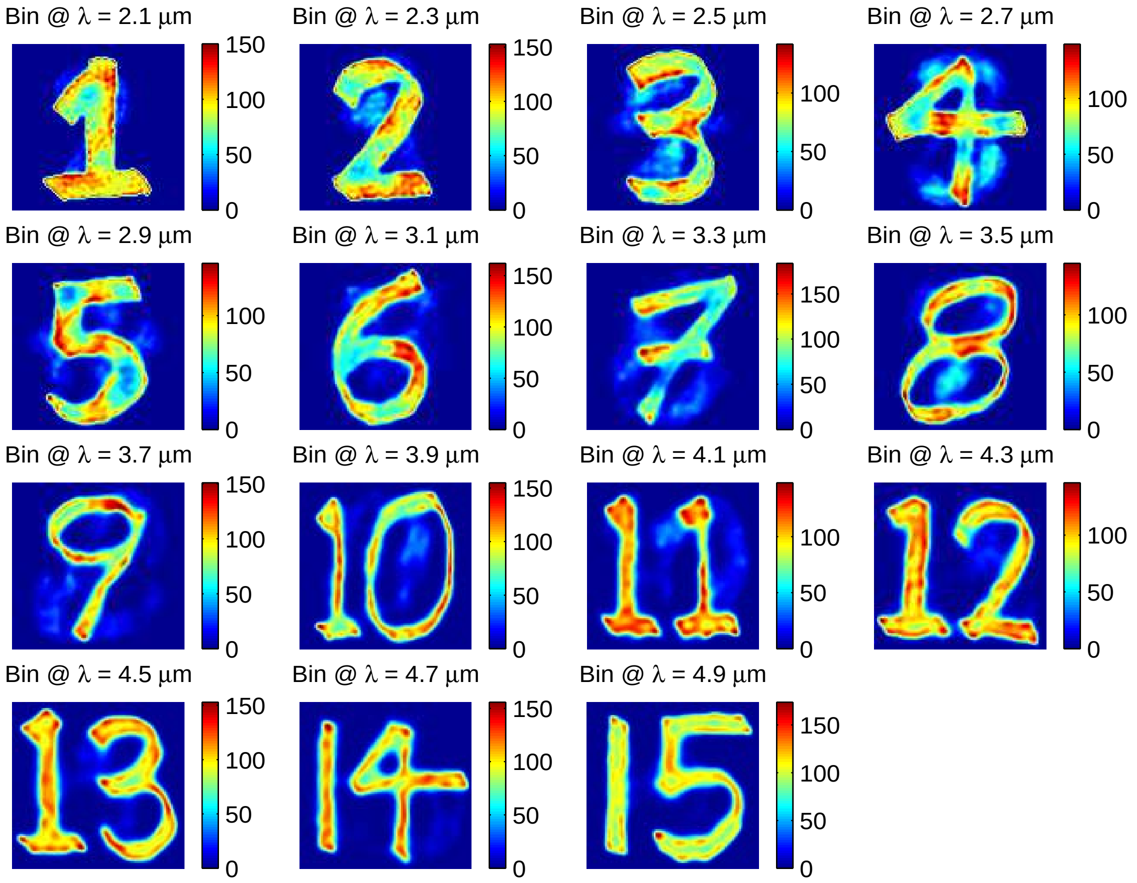


Figure 5.23: This figure shows the reconstruction of the noisy numbers test data cube by the 2D reconstruction algorithm. The reconstruction using noisy data is highly comparable to the previous test case which contained no noise.

Test 4.4.3 uses a detector image which has been attenuated by the atmosphere. The reconstruction is conducted using 100 iterations of the 2D atmospheric reconstruction algorithm. The 2D reconstruction of the attenuated data is highly comparable to the noiseless reconstruction as shown in Figure 5.24 and indicated in Tables 5.15 and 5.16. The exceptions are the $\lambda = 2.7\mu m$ and $\lambda = 4.3\mu m$ bins which fail to reconstruct the source shape due to the low $t_{atm}(\lambda)$ of the bins.

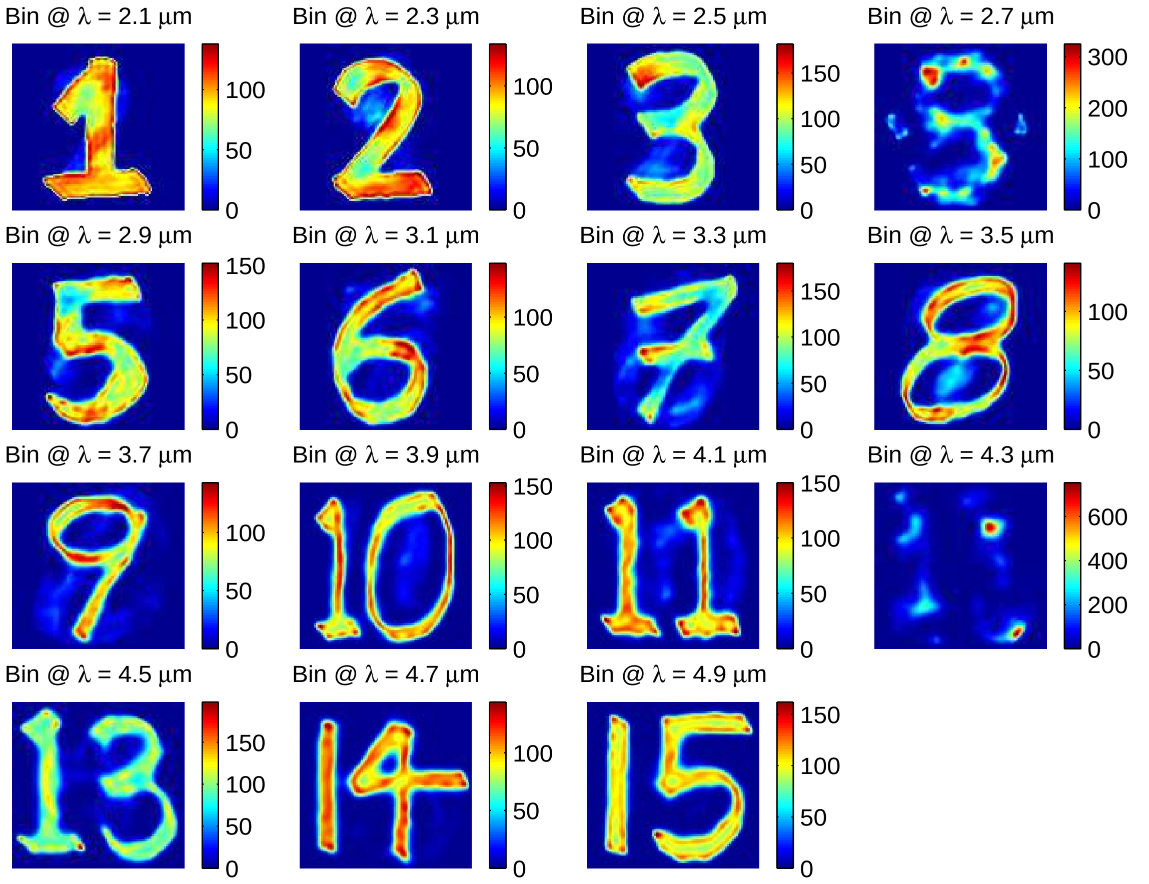


Figure 5.24: This figure shows the reconstruction of the atmospherically attenuated numbers test data cube by the 2D atmospheric reconstruction algorithm. This reconstruction is highly comparable to the noiseless test case with the exception of the $\lambda = 2.7\mu m$ and $\lambda = 4.3\mu m$ bins.

The photon sums in each bin are shown in Figure 5.25 for the numbers test using noiseless, noisy and atmospherically attenuated number test data. The bin photon sums are comparable for the noiseless and noisy data. Most of the bin photon sums of the attenuated data also track the original data.

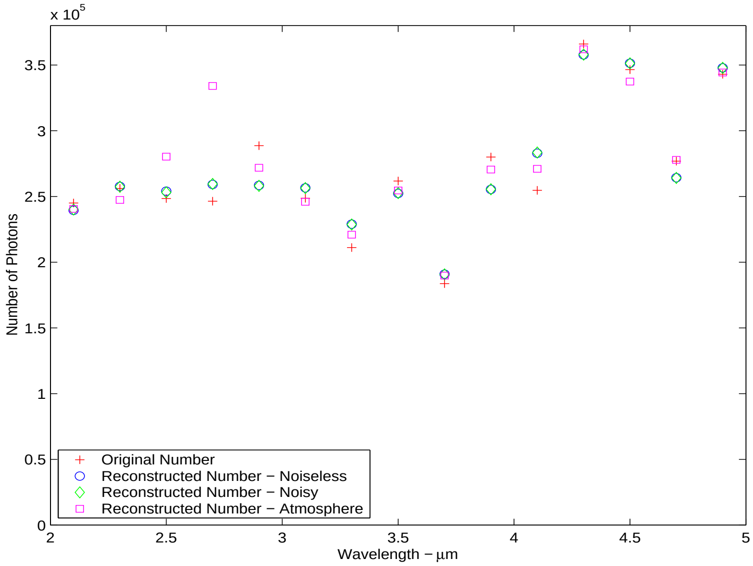


Figure 5.25: This figure shows the sum of photons in each wavelength bin for the numbers test in the reconstructed 2D data cube compared with the original data cube. The sum of photons for the reconstruction of noisy detector data and atmospherically attenuated detector data is also shown. All reconstructed bin photon sums track the original input data.

Further reconstruction performance results the numbers tests in are shown in Tables 5.15 and 5.16. These tables indicate the strong performance of the 2D atmospheric reconstruction algorithm in the bins where the applicable $t_{atm}(\lambda)$ is sufficient to allow some photons to arrive at the detector. Spatial reconstruction of the source shapes degrades with a very low $t_{atm}(\lambda)$, but the bin photon sum is still recovered as indicated by final column in Table 5.15.

Table 5.15: Number Monochromatic Source 2D Reconstruction Photon Sum Results For Clean, Noisy And Atmospherically Attenuated Data

Bin	**Orig**	**Recon**	**Recon**	**Recon**	**R/O**	**R/O**	**R/O**
λ	**Number**	**Number**	**Number N**	**Number A**	**%**	**% N**	**% A**
2.1	245000	239478	240065	240533	97.7	98.0	98.2
2.3	256100	257599	257470	247412	100.6	100.5	96.6
2.5	248500	254051	253153	280336	102.2	101.9	112.8
2.7	246400	259143	259494	334045	105.2	105.3	135.6
2.9	288700	258432	258104	271814	89.5	89.4	94.2
3.1	248600	256577	256379	245967	103.2	103.1	98.9
3.3	211100	228945	228675	220955	108.5	108.3	104.7
3.5	261800	252339	252601	254575	96.4	96.5	97.2
3.7	183700	190862	190705	189807	103.9	103.8	103.3
3.9	280000	255358	255450	270461	91.2	91.2	96.6
4.1	254700	282907	283489	271018	111.1	111.3	106.4
4.3	366000	357764	357890	361925	97.7	97.8	98.9
4.5	346500	351325	351307	337370	101.4	101.4	97.4
4.7	276900	264331	264104	277887	95.5	95.4	100.4
4.9	343000	347891	347866	344364	101.4	101.4	100.4

Table 5.16: Number Monochromatic Source 2D Reconstruction REM Results For Clean, Noisy And Atmospherically Attenuated Data

Bin	**Error**	**Error**	**Error**	**Error**	**Error**	**Error**
λ	**Number**	**Number N**	**Number A**	**%**	**% N**	**% A**
2.1	52499	55270	46205	21.4	22.6	18.9
2.3	82404	84248	70791	32.2	32.9	27.6
2.5	104625	105826	83876	42.1	42.6	33.8
2.7	132353	132507	345454	53.7	53.8	140.2
2.9	110465	111602	84653	38.3	38.7	29.3
3.1	97272	97859	83297	39.1	39.4	33.5
3.3	116760	116899	114047	55.3	55.4	54.0
3.5	113791	114129	104676	43.5	43.6	40.0
3.7	94462	94164	83872	51.4	51.3	45.7
3.9	120647	120390	107284	43.1	43.0	38.3
4.1	107606	107475	100975	42.2	42.2	39.6
4.3	107406	108163	463569	29.3	29.6	126.7
4.5	103747	103957	114087	29.9	30.0	32.9
4.7	95954	96517	84341	34.7	34.9	30.5
4.9	86413	87747	81648	25.2	25.6	23.8

A summary of the results for the monochromatic numbers test cases 4.4.1 to 4.4.3 is presented in Table 5.17. The table again indicates little performance degradation of the 2D algorithm with the addition of noise to the detector image. Atmospherically attenuated detector data is also successfully used to reconstruct the input hyperspectral data cube for the wavelength bins where $t_{atm}(\lambda) > 0.03$. The table indicates reconstruction failure in the reconstruction metrics for the $\lambda = 2.7\mu m$ and $\lambda = 4.3\mu m$ bins where $t_{atm}(\lambda) < 0.03$. Spectral bleeding is also observed in each of the numbers test cases in the form of shadowing of numbers in adjacent bins.

Table 5.17: Number Monochromatic Source Results Summary

Test Case	**Photon Sum Error** %	**REM** %
2D	0.6 - 11.1	21.4 - 55.3
2D + noise	0.5 - 11.3	22.6 - 55.4
2D + atm	1.1 - 12.8	18.9 - 54.0
2D + atm: bad bins	35.6, 6.4	140.2, 126.7

5.6 Static Fireball Results

5.6.1 Uniform Fireball Results. The results for test 4.5.1.1 are obtained by performing 1000 iterations of the vector reconstruction algorithm using a noiseless detector image of the 400^{o}K fireball. The results for test 4.5.1.2 are obtained by performing 1000 iterations of the vector atmospheric reconstruction algorithm with a detector image that is atmospherically attenuated. The original and reconstructed input data cubes for both test cases are shown in Figure 5.26.

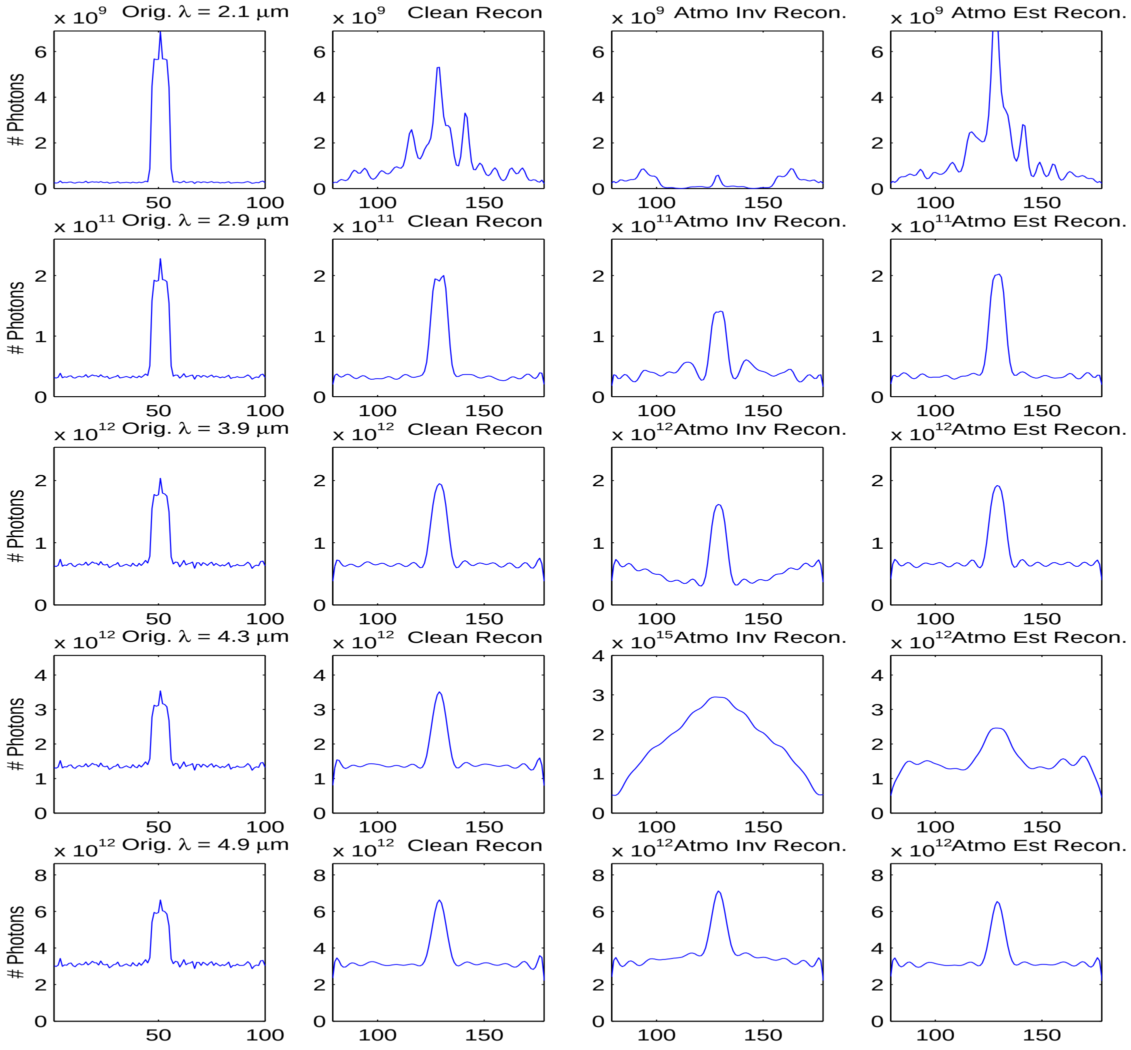


Figure 5.26: This figure shows the reproduction of the original 400^{o}K fireball data cube by the vector reconstruction algorithm. The reconstruction of atmospherically attenuated input data is also shown with reconstruction failure in the $\lambda = 2.1\mu m$ and $\lambda = 4.3\mu m$ bins for the direct atmospheric inverse algorithm. The atmospheric reconstruction is more successful, as shown in the right column, when the atmospheric attenuation data is included within the algorithm.

The vector atmospheric reconstruction algorithm for this test case separately computes results for both the direct atmospheric inverse and the inclusion of the atmospheric data within the estimation algorithm as discussed in section 3.4.3.

The photon sums in each bin are shown in Figure 5.27 demonstrate good tracking of the vector reconstruction algorithm on clean input data. However the error increases with the atmospherically attenuated data and the sum greatly increases in the low valued $t_{atm}(\lambda)$ bins. This is consistent in both vector atmospheric reconstruction algorithms but is more pronounced using the direct atmospheric inverse method. Hence the inclusion of the atmospheric data within the estimation algorithm is regarded as the superior method for reconstructing atmospherically attenuated data.

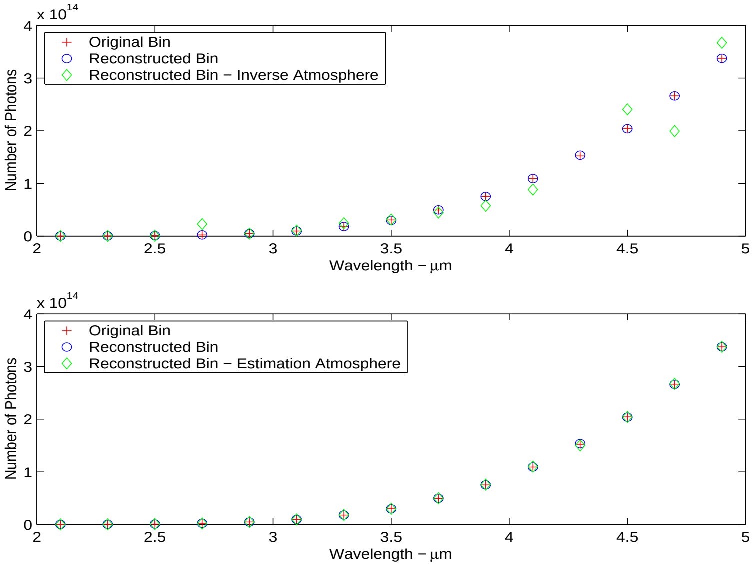


Figure 5.27: This figure shows the sum of photons in each wavelength bin for the 400^{o}K fireball in the reconstructed data compared with the original data cube. The sum of photons for the reconstruction of atmospherically attenuated detector data is also shown for both vector atmospheric reconstruction methods.

The remaining reconstruction results are shown in Tables 5.18 and 5.19. This data shows REM decreasing as the bin wavelength increases. This trend is attributed to Planckian nature of the 400^{o}K blackbody where emitted photon numbers greatly increase between $3 - 5\mu m$.

Table 5.18: Uniform Fireball at 400^{o}K Vector Reconstruction Photon Sum Results For Clean And Atmospherically Attenuated Data

Bin λ	**Orig** **Number**	**Recon** **Number**	**Recon** **Number $\mathbf{A}_{INV}$**	**Recon** **Number $\mathbf{A}_{EST}$**	**R/O** **%**	**R/O** **% $\mathbf{A}_{INV}$**	**R/O** **% $\mathbf{A}_{EST}$**
2.1	7.6E+10	1.2E+11	2.7E+10	1.4E+11	157.1	35.2	182.0
2.3	2.7E+11	3.6E+11	9.8E+10	3.9E+11	132.0	35.9	142.0
2.5	8.2E+11	8.4E+11	3.7E+11	8.7E+11	102.4	44.8	106.0
2.7	2.1E+12	2.2E+12	2.3E+13	2.0E+12	104.1	1087.7	96.0
2.9	4.7E+12	4.7E+12	4.7E+12	4.8E+12	99.3	98.2	102.0
3.1	9.6E+12	9.5E+12	1.0E+13	9.4E+12	98.5	108.4	98.0
3.3	1.8E+13	1.8E+13	2.4E+13	1.8E+13	101.2	136.4	101.0
3.5	3.1E+13	3.0E+13	3.1E+13	3.0E+13	97.0	101.9	97.2
3.7	4.9E+13	5.0E+13	4.5E+13	5.0E+13	100.5	90.2	101.0
3.9	7.5E+13	7.5E+13	5.8E+13	7.6E+13	100.0	76.6	101.0
4.1	1.1E+14	1.1E+14	8.8E+13	1.1E+14	99.9	80.8	101.0
4.3	1.5E+14	1.5E+14	1.8E+17	1.5E+14	100.9	119900.9	98.7
4.5	2.0E+14	2.0E+14	2.4E+14	2.0E+14	99.6	117.7	99.8
4.7	2.7E+14	2.7E+14	2.0E+14	2.7E+14	100.0	74.9	100.0
4.9	3.4E+14	3.4E+14	3.7E+14	3.4E+14	100.0	108.8	100.0

Table 5.19: Uniform Fireball at 400^{o}K Vector Reconstruction REM Results For Clean And Atmospherically Attenuated Data

Bin λ	**Error** **Number**	**Error** **Number $\mathbf{A}_{INV}$**	**Error** **Number $\mathbf{A}_{EST}$**	**Error** **%**	**Error** **% $\mathbf{A}_{INV}$**	**Error** **% $\mathbf{A}_{EST}$**
2.1	7.8E+10	6.8E+10	7.5E+10	102.8	89.8	99.9
2.3	1.3E+11	1.9E+11	1.2E+11	46.9	67.8	44.8
2.5	1.9E+11	4.6E+11	2.1E+11	23.8	56.1	25.9
2.7	2.7E+11	2.1E+13	6.6E+11	12.7	985.9	31.3
2.9	4.9E+11	1.3E+12	5.3E+11	10.4	27.9	11.2
3.1	8.7E+11	1.4E+12	9.3E+11	9.0	14.9	9.6
3.3	1.4E+12	6.8E+12	1.4E+12	7.6	38.1	7.9
3.5	2.3E+12	3.1E+12	2.2E+12	7.5	10.2	7.3
3.7	3.1E+12	6.3E+12	3.3E+12	6.4	12.8	6.7
3.9	4.7E+12	1.8E+13	4.8E+12	6.3	24.5	6.4
4.1	6.3E+12	2.2E+13	6.5E+12	5.8	19.9	5.9
4.3	8.5E+12	1.8E+17	2.3E+13	5.6	118648.8	15.4
4.5	1.1E+13	4.1E+13	1.2E+13	5.6	20.2	5.7
4.7	1.4E+13	7.0E+13	1.4E+13	5.4	26.3	5.3
4.9	1.5E+13	3.4E+13	1.5E+13	4.4	10.0	4.4

The results for test 4.5.1.3 are obtained by performing 1000 iterations of the vector reconstruction algorithm using a noiseless detector image of the 1000^{o}K fireball. The results for test 4.5.1.4 are obtained by performing 1000 iterations of the vector atmospheric reconstruction algorithm with a detector image that is atmospherically attenuated. The original and reconstructed input data cubes for both test cases are shown in Figure 5.28.

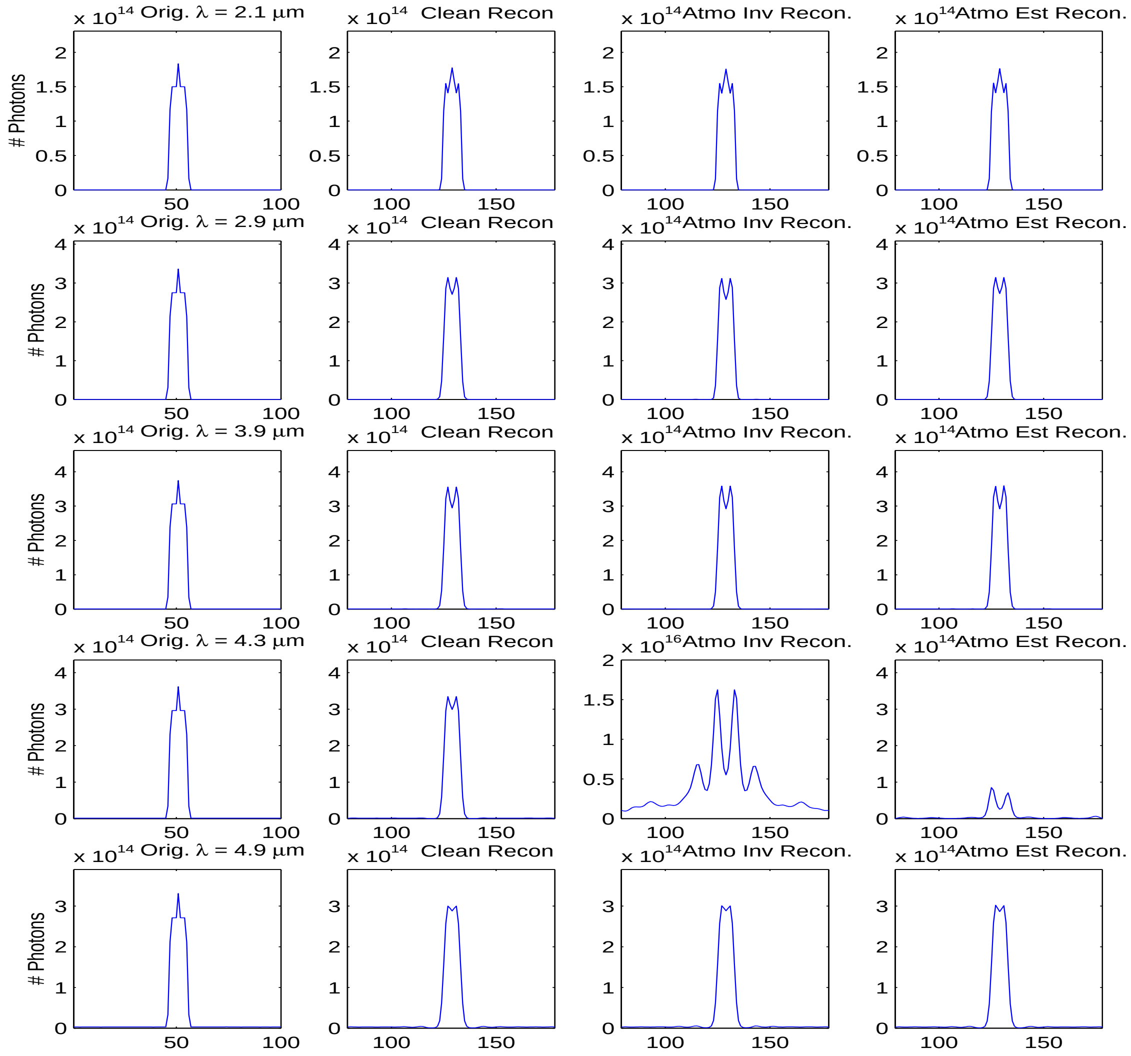


Figure 5.28: This figure shows the reproduction of the original 1000^{o}K fireball data cube by the vector reconstruction algorithm. The reconstruction of atmospherically attenuated input data is also shown with reconstruction failure in the $\lambda = 4.3\mu m$ bin more severe in the direct atmospheric inverse method.

The photon sums in each bin are shown in Figure 5.29 demonstrate good tracking of the vector reconstruction algorithm on clean input data. However, as with the previous test case, the error increases with the atmospherically attenuated data and the sum greatly increases in the low valued $t_{atm}(\lambda)$ bins. Again the direct atmospheric inverse algorithm performs more poorly than the inclusion of atmospheric data within the estimation algorithm.

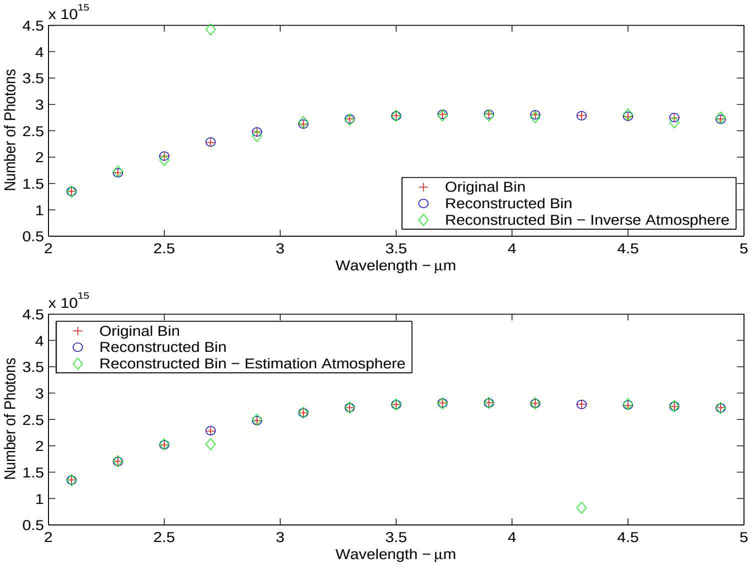


Figure 5.29: This figure shows the sum of photons in each wavelength bin for the 1000^{o}K fireball in the reconstructed data compared with the original data cube. The sum of photons for the reconstruction of atmospherically attenuated detector data is also shown for both atmospheric reconstruction methods.

The remaining reconstruction results are shown in Tables 5.20 and 5.21. The data indicates little drop-off in reconstruction performance when the atmospheric attenuation is included. The exception is again the low valued $t_{atm}(\lambda)$ bins at $\lambda = 2.7\mu m$ and $\lambda = 4.3\mu m$, which is more severe for the direct atmospheric inversion method.

Table 5.20: Uniform Fireball at 1000^{o}K Vector Reconstruction Photon Sum Results For Clean And Atmospherically Attenuated Data

Bin λ	**Orig** **Number**	**Recon** **Number**	**Recon** **Number** $\mathbf{A}_{INV}$	**Recon** **Number** $\mathbf{A}_{EST}$	**R/O** **%**	**R/O** **%** $\mathbf{A}_{INV}$	**R/O** **%** $\mathbf{A}_{EST}$
2.1	1.4E+15	1.3E+15	1.3E+15	1.3E+15	99.8	99.4	99.7
2.3	1.7E+15	1.7E+15	1.7E+15	1.7E+15	99.8	101.4	100.1
2.5	2.0E+15	2.0E+15	1.9E+15	2.0E+15	100.1	96.3	100.5
2.7	2.3E+15	2.3E+15	4.4E+15	2.0E+15	100.4	194.2	89.2
2.9	2.5E+15	2.5E+15	2.4E+15	2.5E+15	99.9	96.8	100.5
3.1	2.6E+15	2.6E+15	2.7E+15	2.6E+15	100.1	101.5	100.0
3.3	2.7E+15	2.7E+15	2.7E+15	2.7E+15	100.0	99.4	100.0
3.5	2.8E+15	2.8E+15	2.8E+15	2.8E+15	100.0	100.0	100.1
3.7	2.8E+15	2.8E+15	2.8E+15	2.8E+15	100.0	99.3	99.8
3.9	2.8E+15	2.8E+15	2.8E+15	2.8E+15	99.8	99.1	99.8
4.1	2.8E+15	2.8E+15	2.8E+15	2.8E+15	99.8	98.1	100.0
4.3	2.8E+15	2.8E+15	3.7E+17	8.2E+14	99.9	13424.9	29.6
4.5	2.8E+15	2.8E+15	2.8E+15	2.8E+15	100.4	101.6	100.9
4.7	2.7E+15	2.8E+15	2.7E+15	2.8E+15	100.3	97.0	100.2
4.9	2.7E+15	2.7E+15	2.8E+15	2.7E+15	99.7	101.0	99.7

Table 5.21: Uniform Fireball at 1000^{o}K Vector Reconstruction REM Results For Clean And Atmospherically Attenuated Data

Bin λ	**Error** **Number**	**Error** **Number** $\mathbf{A}_{INV}$	**Error** **Number** $\mathbf{A}_{EST}$	**Error** **%**	**Error** **%** $\mathbf{A}_{INV}$	**Error** **%** $\mathbf{A}_{EST}$
2.1	5.5E+13	5.6E+13	5.7E+13	4.1	4.1	4.2
2.3	9.6E+13	9.3E+13	9.4E+13	5.6	5.4	5.5
2.5	1.7E+14	1.2E+14	1.7E+14	8.3	5.8	8.3
2.7	2.7E+14	2.1E+15	4.9E+14	11.9	94.2	21.4
2.9	3.4E+14	3.2E+14	3.4E+14	13.7	13.1	13.7
3.1	4.0E+14	4.0E+14	3.8E+14	15.3	15.4	14.4
3.3	4.3E+14	4.3E+14	4.1E+14	15.7	15.8	15.0
3.5	4.3E+14	4.3E+14	4.1E+14	15.3	15.4	14.8
3.7	4.3E+14	4.4E+14	4.2E+14	15.3	15.8	14.8
3.9	4.2E+14	4.4E+14	4.3E+14	14.9	15.6	15.1
4.1	4.0E+14	4.1E+14	4.1E+14	14.3	14.5	14.7
4.3	3.9E+14	3.7E+17	2.3E+15	13.9	13227.8	83.5
4.5	4.2E+14	4.9E+14	4.3E+14	15.2	17.8	15.4
4.7	4.4E+14	4.4E+14	4.2E+14	16.2	16.1	15.2
4.9	4.2E+14	4.3E+14	4.2E+14	15.5	15.8	15.3

The results for test 4.5.1.5 are obtained by performing 1000 iterations of the vector reconstruction algorithm using a noiseless detector image of the 1600^{o}K fireball. The results for test 4.5.1.6 are obtained by performing 1000 iterations of the vector atmospheric reconstruction algorithm with a detector image that is atmospherically attenuated. The original and reconstructed input data cubes for both test cases are shown in Figure 5.30.

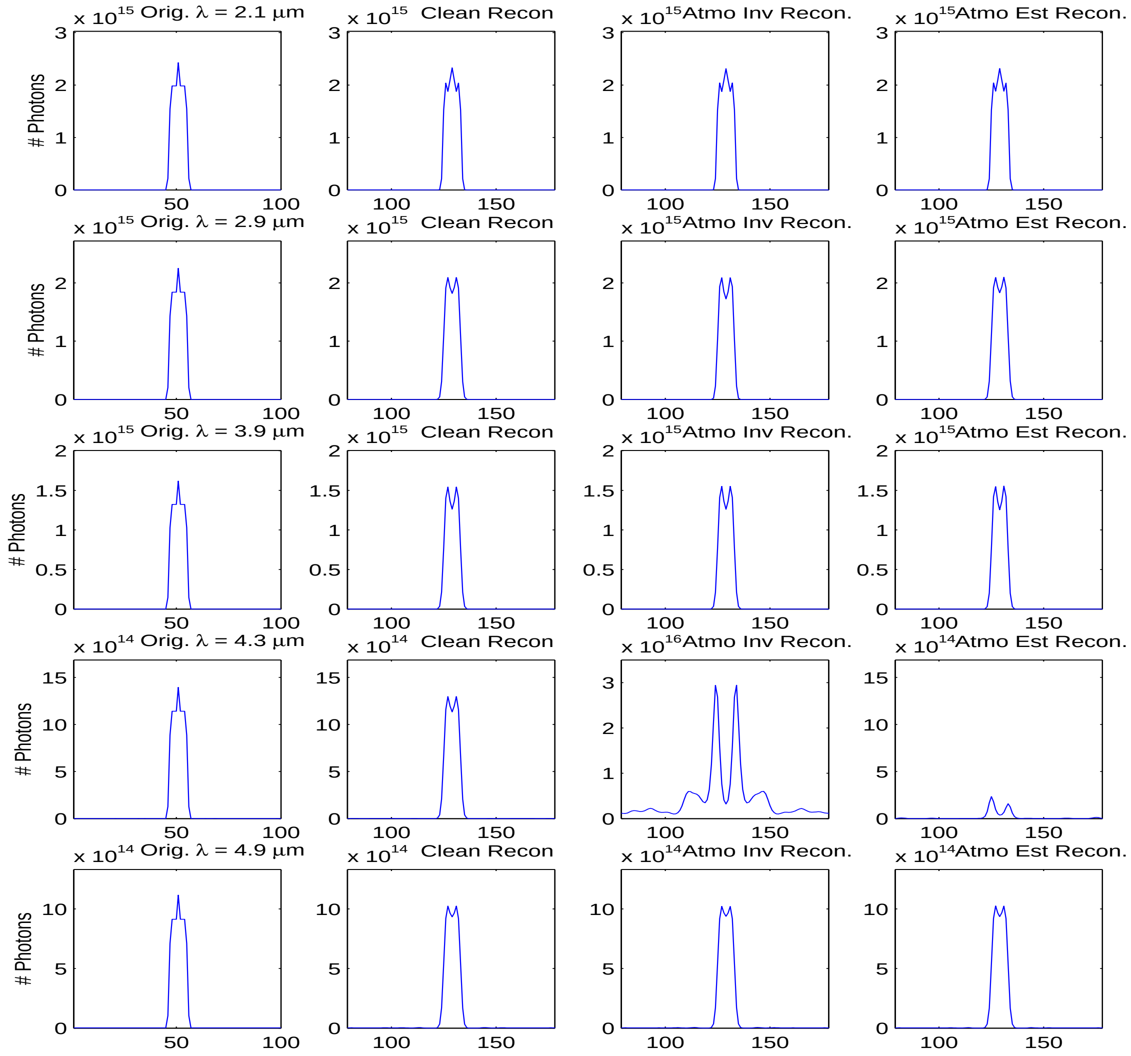


Figure 5.30: This figure shows the reproduction of the original 1600^{o}K fireball data cube by the vector reconstruction algorithm. The reconstruction of atmospherically attenuated input data is also shown with reconstruction failure in the $\lambda = 4.3\mu m$ bin for both atmospheric reconstruction methods.

The photon sums in each bin are shown in Figure 5.31 and demonstrates good tracking of the vector reconstruction algorithms on both clean and atmospherically attenuated input data. However as with the previous test cases for the atmospheric reconstruction, the error in the sum greatly increases in the low valued $t_{atm}(\lambda)$ bins. The inclusion of the atmospheric data with the algorithm again shows better reconstruction performance.

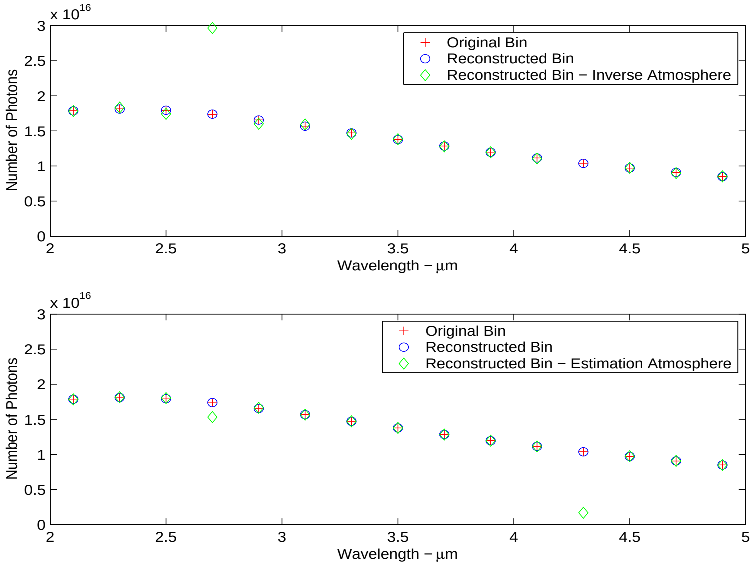


Figure 5.31: This figure shows the sum of photons in each wavelength bin for the 1600^{o}K fireball in the reconstructed data compared with the original data cube. The sum of photons for the reconstruction of atmospherically attenuated detector data is also shown.

The remaining reconstruction results are shown in Tables 5.22 and 5.23. Again the data indicates little drop-off in reconstruction performance when the atmospheric attenuation is included. However as previously stated the exception is the low valued $t_{atm}(\lambda)$ bins at $\lambda = 2.7\mu m$ and $\lambda = 4.3\mu m$.

Table 5.22: Uniform Fireball at 1600^oK Vector Reconstruction Photon Sum Results For Clean And Atmospherically Attenuated Data

Bin	**Orig**	**Recon**	**Recon**	**Recon**	**R/O**	**R/O**	**R/O**
λ	**Number**	**Number**	**Number $\mathbf{A}_{INV}$**	**Number $\mathbf{A}_{EST}$**	**%**	**% $\mathbf{A}_{INV}$**	**% $\mathbf{A}_{EST}$**
2.1	1.8E+16	1.8E+16	1.8E+16	1.8E+16	99.9	99.7	99.8
2.3	1.8E+16	1.8E+16	1.8E+16	1.8E+16	99.8	101.0	100.0
2.5	1.8E+16	1.8E+16	1.7E+16	1.8E+16	100.2	97.3	100.6
2.7	1.7E+16	1.7E+16	3.0E+16	1.5E+16	100.3	170.9	88.3
2.9	1.7E+16	1.7E+16	1.6E+16	1.7E+16	99.9	96.7	100.4
3.1	1.6E+16	1.6E+16	1.6E+16	1.6E+16	100.0	101.6	100.0
3.3	1.5E+16	1.5E+16	1.5E+16	1.5E+16	99.9	99.0	99.9
3.5	1.4E+16	1.4E+16	1.4E+16	1.4E+16	100.0	100.3	100.0
3.7	1.3E+16	1.3E+16	1.3E+16	1.3E+16	100.0	99.6	99.9
3.9	1.2E+16	1.2E+16	1.2E+16	1.2E+16	100.0	99.9	99.9
4.1	1.1E+16	1.1E+16	1.1E+16	1.1E+16	100.0	99.2	100.0
4.3	1.0E+16	1.0E+16	4.6E+17	1.7E+15	99.9	4456.4	16.3
4.5	9.7E+15	9.7E+15	9.7E+15	9.7E+15	100.2	100.3	100.5
4.7	9.1E+15	9.1E+15	9.0E+15	9.1E+15	100.1	99.2	100.1
4.9	8.5E+15	8.5E+15	8.5E+15	8.5E+15	99.8	100.2	99.9

A summary of the results for the uniform fireball test cases 4.5.1.1 to 4.5.1.6 is presented in Table 5.24. The table indicates the better reconstruction performance obtained when including the atmospheric data with the reconstruction algorithm for the 1000^oK and 1600^oK fireballs. The apparent discrepancy with the 400^oK fireball is again attributed to the Planckian nature of the 400^oK blackbody, where emitted photon numbers greatly increase between $3 - 5\mu m$.

The temperatures of each uniform fireball are also estimated from the reconstructed photon data for each test case as shown in Table 5.25. The table shows the estimate obtained by each atmospheric reconstruction algorithm method. The temperature estimates obtained from the direct atmospheric inverse method are all higher than the nominal fireball temperature and show greater error due to the low valued $t_{atm}(\lambda)$ bins. The reconstructed temperature is more accurate when the atmospheric data is included within the reconstruction algorithm. However, when a mask is applied to remove the low valued $t_{atm}(\lambda)$ bins, both atmospherically reconstructed temperatures almost match the reconstructed temperature from the pristine detector data case.

Table 5.23: Uniform Fireball at 1600^{o}K Vector Reconstruction REM Results For Clean And Atmospherically Attenuated Data

Bin	**Error**	**Error**	**Error**	**Error**	**Error**	**Error**
λ	**Number**	**Number A$_{INV}$**	**Number A$_{EST}$**	**%**	**% A$_{INV}$**	**% A$_{EST}$**
2.1	6.9E+14	6.9E+14	6.9E+14	3.9	3.9	3.9
2.3	9.5E+14	9.4E+14	9.3E+14	5.3	5.2	5.1
2.5	1.4E+15	9.8E+14	1.4E+15	7.8	5.5	7.9
2.7	1.9E+15	1.2E+16	3.7E+15	11.2	70.9	21.2
2.9	2.2E+15	2.1E+15	2.2E+15	13.4	12.9	13.5
3.1	2.4E+15	2.4E+15	2.2E+15	15.1	15.0	14.1
3.3	2.3E+15	2.3E+15	2.2E+15	15.6	15.5	14.9
3.5	2.1E+15	2.1E+15	2.0E+15	15.2	15.3	14.7
3.7	2.0E+15	2.0E+15	1.9E+15	15.2	15.4	14.7
3.9	1.8E+15	1.8E+15	1.8E+15	14.9	15.2	15.0
4.1	1.6E+15	1.6E+15	1.6E+15	14.2	14.3	14.4
4.3	1.4E+15	4.5E+17	9.4E+15	13.8	4330.5	90.6
4.5	1.4E+15	1.4E+15	1.4E+15	14.1	14.6	14.0
4.7	1.3E+15	1.3E+15	1.2E+15	14.6	14.3	13.8
4.9	1.1E+15	1.1E+15	1.1E+15	13.1	13.1	13.1

Table 5.24: Uniform Fireballs Results Summary

	Photon Sum Error	**REM**
Test Case	**%**	**%**
1D (400^{o}K)	0.0 - 57.1	4.4 - 102.8
1D + atm (post-inverse)	1.9 - 64.8	10.0 - 89.8
1D + atm (estimation)	0.0 - 82.0	4.4 - 99.9
1D (1000^{o}K)	0.0 - 0.4	4.1 - 16.2
1D + atm (post-inverse)	0.0 - 3.2	4.1 - 17.8
1D + atm (estimation)	0.0 - 0.9	4.2 - 15.4
1D (1600^{o}K)	0.0 - 0.3	3.9 - 15.6
1D + atm (post-inverse)	0.3 - 3.3	3.9 - 15.5
1D + atm (estimation)	0.0 - 0.6	3.9 - 15.0

Table 5.25: Uniform Fireballs Reconstructed Temperatures

	Reconstructed Fireball Temperature			
Test Case	**All Bins**	**Error %**	**With Mask**	**Error %**
1D (400K)	424	6.0		
1D + atm (post-inverse)	904	126.0	423	5.8
1D + atm (estimation)	422	5.5	424	6.0
1D (1000K)	1003	0.3		
1D + atm (post-inverse)	1234	23.4	1003	0.3
1D + atm (estimation)	987	-1.3	1003	0.3
1D (1600K)	1586	-0.9		
1D + atm (post-inverse)	1652	3.3	1582	-1.1
1D + atm (estimation)	1566	-2.1	1584	-1.0

5.6.2 Concentric Temperature Rings Fireball Results. The results for test 4.5.2.1 are obtained by performing 100 iterations of the 2D reconstruction algorithm using a noiseless detector image of the concentric rings fireball. The results for test 4.5.2.2 are are obtained by performing 100 iterations of the 2D atmospheric reconstruction algorithm with a detector image that is atmospherically attenuated. The original and reconstructed input data cubes for both test cases are shown in Figure 5.32.

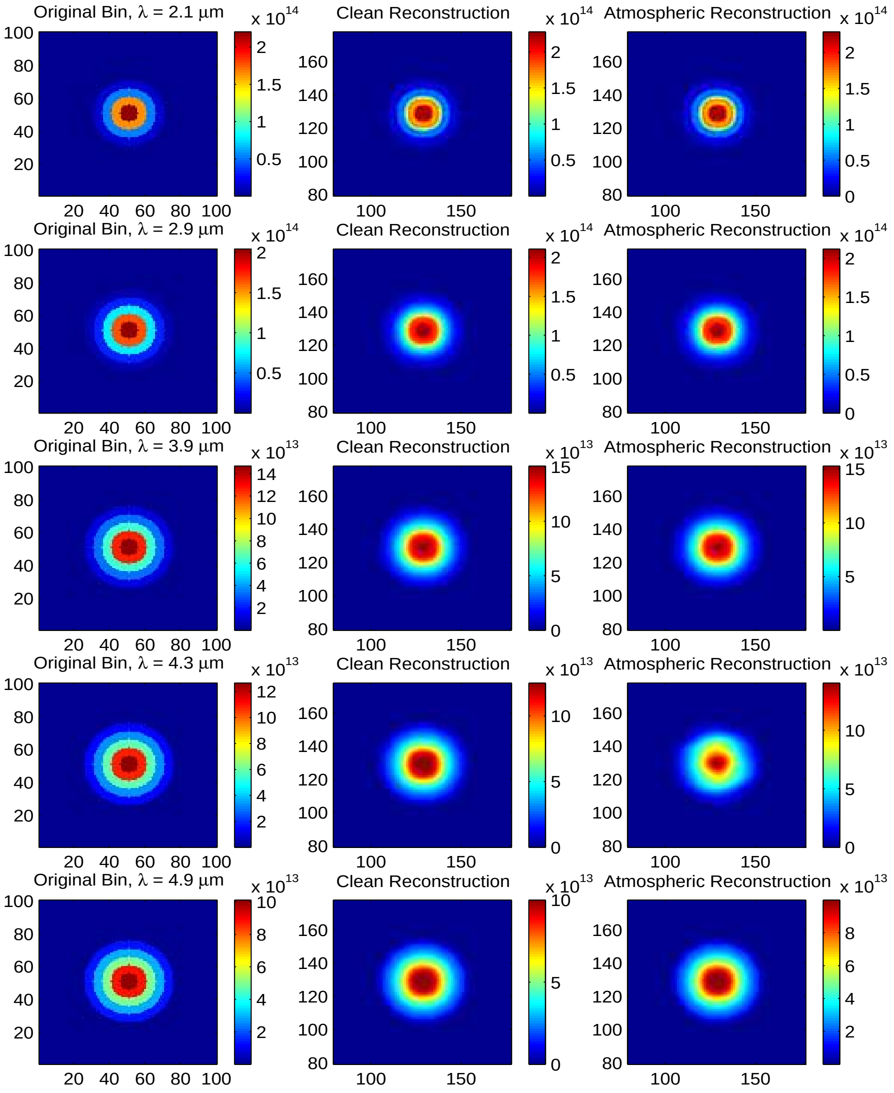


Figure 5.32: This figure shows the reproduction of the original concentric rings fireball data cube by both the 2D reconstruction algorithm and 2D atmospheric reconstruction algorithm. Note that reconstruction of the atmospherically attenuated input data in the $\lambda = 4.3\mu m$ bin is achieved for this data set.

The photon sums in each bin are shown in Figure 5.33 and demonstrate good tracking of the 2D reconstruction algorithms on both clean and atmospherically attenuated input data. These results also show better reconstruction of the low valued $t_{atm}(\lambda)$ bins possibly due to the high number photons in the original data cube.

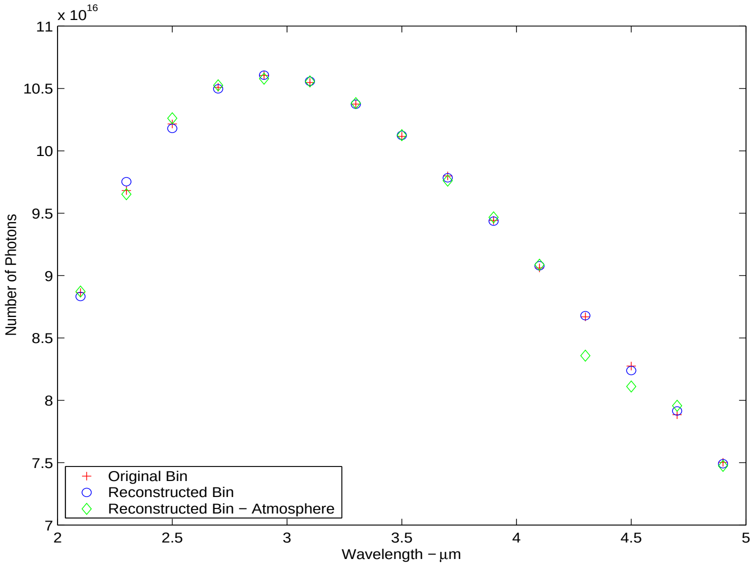


Figure 5.33: This figure shows the sum of photons in each wavelength bin for the concentric temperature rings fireball in the reconstructed 2D data cube compared with the original data cube. The sum of photons for the reconstruction of atmospherically attenuated detector data is also shown.

The remaining reconstruction results are shown in Table 5.26. Again the data indicates little drop-off in reconstruction performance when the atmospheric attenuation is included and also shows good REM performance for the low valued $t_{atm}(\lambda)$ bins.

Table 5.26: Concentric Temperature Rings Fireball 2D Reconstruction Results For Clean And Atmospherically Attenuated Data

Bin	**Orig**	**Recon**	**Recon**	**R/O**	**R/O**	**Error**	**Error**	**Error**	**Error**
λ	**Bin**	**Bin**	**Bin A**	**%**	**% A**	**Photons**	**Photons A**	**%**	**% A**
2.1	8.9E+16	8.8E+16	8.9E+16	99.6	100.1	9.3E+15	9.3E+15	10.5	10.5
2.3	9.7E+16	9.8E+16	9.7E+16	100.7	99.7	1.2E+16	1.1E+16	12.1	11.4
2.5	1.0E+17	1.0E+17	1.0E+17	99.6	100.4	1.5E+16	1.5E+16	14.2	14.8
2.7	1.1E+17	1.0E+17	1.1E+17	99.9	100.2	1.6E+16	2.1E+16	15.2	19.7
2.9	1.1E+17	1.1E+17	1.1E+17	100.0	99.7	1.7E+16	1.7E+16	16.0	16.2
3.1	1.1E+17	1.1E+17	1.1E+17	100.1	100.1	1.8E+16	1.7E+16	16.7	16.6
3.3	1.0E+17	1.0E+17	1.0E+17	100.0	100.1	1.7E+16	1.7E+16	16.8	16.8
3.5	1.0E+17	1.0E+17	1.0E+17	100.1	100.1	1.7E+16	1.7E+16	16.4	16.4
3.7	9.8E+16	9.8E+16	9.8E+16	99.8	99.6	1.6E+16	1.6E+16	16.1	16.1
3.9	9.4E+16	9.4E+16	9.5E+16	99.9	100.3	1.5E+16	1.5E+16	15.9	15.8
4.1	9.1E+16	9.1E+16	9.1E+16	100.2	100.3	1.4E+16	1.4E+16	15.7	15.7
4.3	8.7E+16	8.7E+16	8.4E+16	100.1	96.4	1.3E+16	1.7E+16	15.5	19.3
4.5	8.3E+16	8.2E+16	8.1E+16	99.6	98.0	1.3E+16	1.3E+16	15.4	15.7
4.7	7.9E+16	7.9E+16	8.0E+16	100.4	100.9	1.2E+16	1.2E+16	15.3	15.3
4.9	7.5E+16	7.5E+16	7.5E+16	99.9	99.6	1.1E+16	1.1E+16	14.9	14.9

The results for test 4.5.2.3 are obtained by performing 1000 iterations of the vector reconstruction algorithm using a noiseless detector image of the concentric rings fireball. The results for test 4.5.2.4 are are obtained by performing 1000 iterations of the vector atmospheric reconstruction algorithm with a detector image that is atmospherically attenuated. The original and reconstructed input data cubes for both test cases are shown in Figure 5.34.

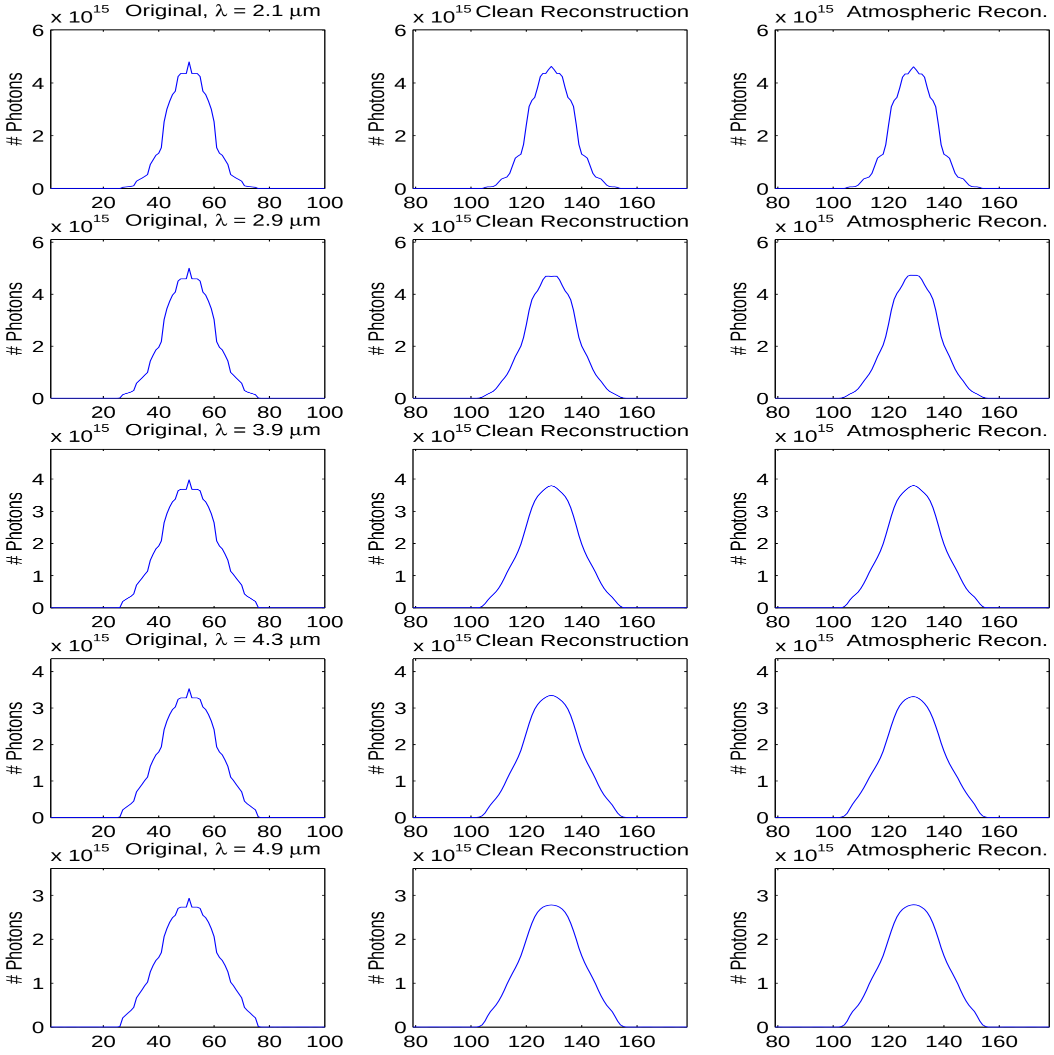


Figure 5.34: This figure shows the reproduction of the original concentric rings fireball data cube by the vector reconstruction algorithm. The reconstruction of atmospherically attenuated input data is also shows good performance across all bins using the estimated atmospheric inverse.

The photon sums in each bin are shown in Figure 5.35 and demonstrate good tracking of the vector reconstruction algorithms on the clean input data. Reconstruction of the photon sums for the atmospherically attenuated data also shows good performance across all wavelength bins.

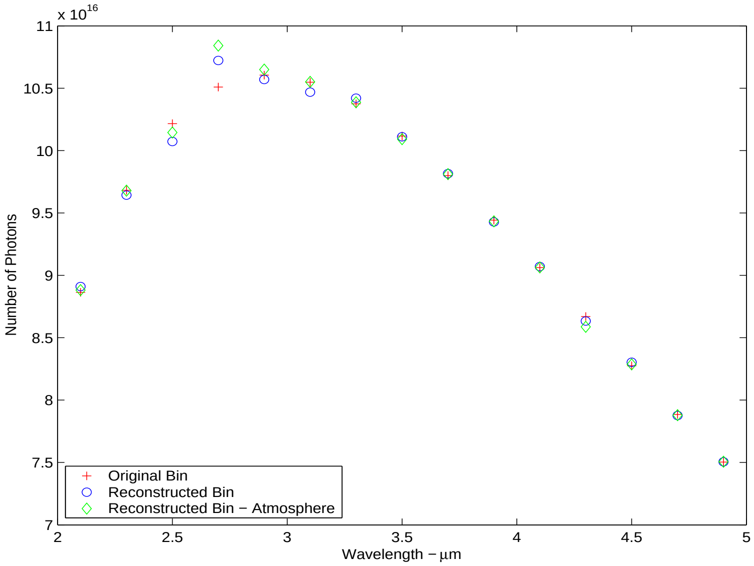


Figure 5.35: This figure shows the sum of photons in each wavelength bin for the concentric rings fireball in the reconstructed data compared with the original data cube. The sum of photons for the reconstruction of atmospherically attenuated detector data is also shown.

The remaining reconstruction results are shown in Table 5.27. The data indicates little drop-off in reconstruction performance across all wavelength bins when the atmospheric attenuation is included.

A summary of the results for the concentric temperature rings fireball test cases 4.5.2.1 to 4.5.2.4 is presented in Table 5.28. The table indicates only a slight performance degradation when atmospherically attenuated detector data is used in both the 2D and vector reconstruction algorithms.

The temperatures of each ring and the background are also estimated from the reconstructed photon data for each of the 2D test cases as shown in Table 5.29. The estimated temperature is accurate to within 3% for each of the reconstructed rings and the background.

Table 5.27: Concentric Temperature Rings Fireball Vector Reconstruction Results For Clean And Atmospherically Attenuated Data

Bin λ	**Orig** **Bin**	**Recon** **Bin**	**Recon** **Bin A**	**R/O** **%**	**R/O** **% A**	**Error** **Photons**	**Error** **Photons A**	**Error** **%**	**Error** **% A**
2.1	8.9E+16	8.9E+16	8.9E+16	100.5	100.2	2.4E+15	2.5E+15	2.7	2.8
2.3	9.7E+16	9.6E+16	9.7E+16	99.6	100.0	2.6E+15	2.6E+15	2.7	2.7
2.5	1.0E+17	1.0E+17	1.0E+17	98.6	99.3	2.9E+15	2.7E+15	2.8	2.7
2.7	1.1E+17	1.1E+17	1.1E+17	102.0	103.2	4.1E+15	5.1E+15	3.9	4.9
2.9	1.1E+17	1.1E+17	1.1E+17	99.7	100.4	3.6E+15	3.9E+15	3.4	3.6
3.1	1.1E+17	1.0E+17	1.1E+17	99.2	100.0	4.1E+15	3.9E+15	3.9	3.7
3.3	1.0E+17	1.0E+17	1.0E+17	100.4	100.1	4.1E+15	3.9E+15	3.9	3.7
3.5	1.0E+17	1.0E+17	1.0E+17	99.9	99.8	3.8E+15	3.8E+15	3.8	3.8
3.7	9.8E+16	9.8E+16	9.8E+16	100.2	100.1	3.7E+15	3.6E+15	3.7	3.7
3.9	9.4E+16	9.4E+16	9.4E+16	99.9	99.9	3.5E+15	3.4E+15	3.7	3.6
4.1	9.1E+16	9.1E+16	9.1E+16	100.1	100.0	3.2E+15	3.2E+15	3.5	3.5
4.3	8.7E+16	8.6E+16	8.6E+16	99.6	99.1	3.0E+15	3.5E+15	3.5	4.1
4.5	8.3E+16	8.3E+16	8.3E+16	100.3	100.1	2.9E+15	2.9E+15	3.5	3.5
4.7	7.9E+16	7.9E+16	7.9E+16	99.9	99.9	2.7E+15	2.7E+15	3.4	3.4
4.9	7.5E+16	7.5E+16	7.5E+16	100.0	100.0	2.4E+15	2.4E+15	3.2	3.2

Table 5.28: Concentric Temperature Rings Fireball Results Summary

Test Case	**Photon Sum Error** **%**	**REM** **%**
2D	0 - 0.7	10.5 - 16.8
2D + atm	0.1 - 3.6	10.5 - 19.7
1D	0.0 - 2.0	2.7 - 3.9
1D + atm	0.0 - 3.2	2.7 - 4.9

Table 5.29: Concentric Temperature Rings Fireball Reconstructed Temperatures from 2D Reconstruction Algorithm

Original **Temperature Ring** **K**	**Reconstructed Fireball Temperature**			
	Clean Data **Estimate** **K**	**Error** **%**	**Atmospheric Data** **Estimate** **K**	**Error** **%**
1600	1602	0.1	1605	0.3
1500	1499	-0.1	1499	-0.1
1200	1203	0.3	1200	0.0
1000	981	-1.9	983	-1.7
800	819	2.4	823	2.9
300	299	-0.3	296	-1.3

5.6.3 Hot-Spot Fireball Results. The results for test 4.5.3.1 are obtained by performing 100 iterations of the 2D reconstruction algorithm using a noiseless detector image of the hot-spots fireball. The results for test 4.5.3.2 are are obtained by performing 100 iterations of the 2D atmospheric reconstruction algorithm with a detector image that is atmospherically attenuated. The original and reconstructed input data cubes for both test cases are shown in Figure 5.36.

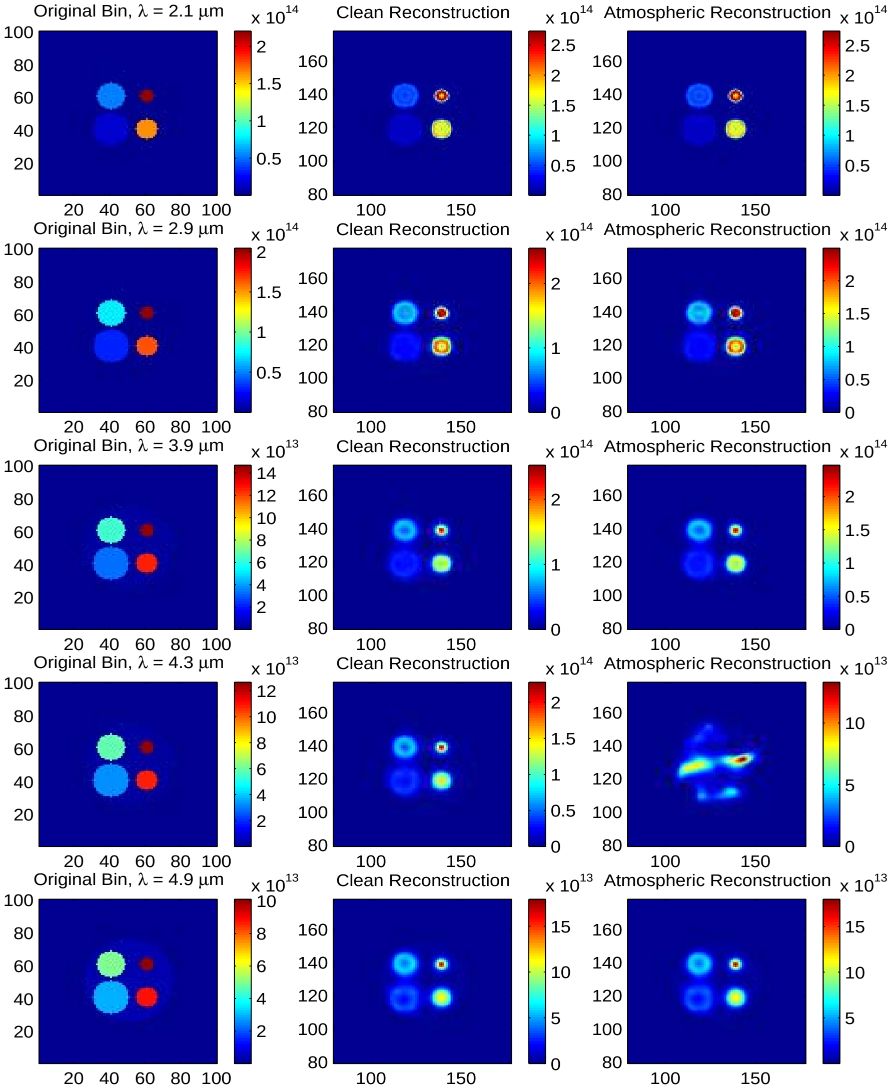


Figure 5.36: This figure shows the reproduction of the original hot-spots rings fireball data cube by both the 2D reconstruction algorithm and 2D atmospheric reconstruction algorithm. The reconstruction of the atmospherically attenuated input data in the $\lambda = 4.3\mu m$ bin is not achieved for this data set.

The photon sums in each bin are shown in Figure 5.37 and demonstrate good tracking of the 2D reconstruction algorithms on both clean and atmospherically attenuated input data. Once again the deviation is greatest in the low valued $t_{atm}(\lambda)$ bins.

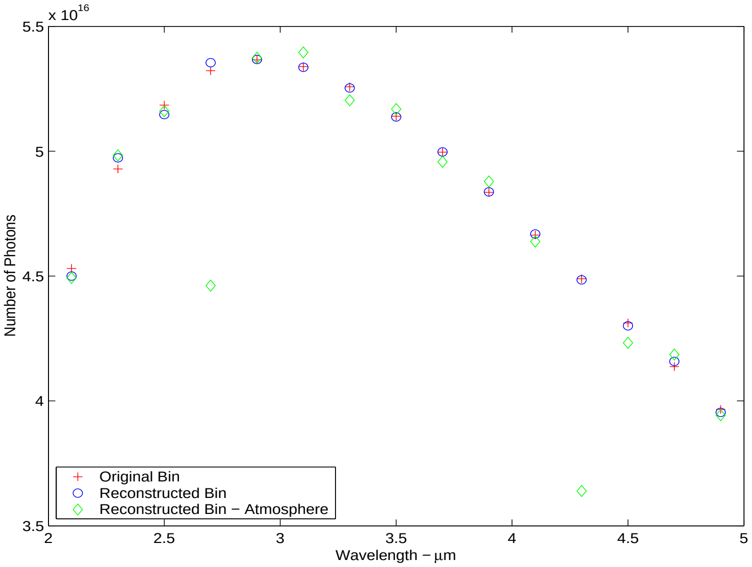


Figure 5.37: This figure shows the sum of photons in each wavelength bin for the hot-spots fireball in the reconstructed 2D data cube compared with the original data cube. The sum of photons for the reconstruction of atmospherically attenuated detector data is also shown.

The remaining reconstruction results are shown in Table 5.30. Again the data indicates little drop-off in reconstruction performance when the atmospheric attenuation is included and also shows degraded REM performance for the low valued $t_{atm}(\lambda)$ bins.

Table 5.30: Hot-Spots Fireball 2D Reconstruction Results For Clean And Atmospherically Attenuated Data

Bin λ	**Orig Bin**	**Recon Bin**	**Recon Bin A**	**R/O %**	**R/O % A**	**Error Photons**	**Error Photons A**	**Error %**	**Error % A**
2.1	4.5E+16	4.5E+16	4.5E+16	99.3	99.2	1.0E+16	9.8E+15	22.1	21.7
2.3	4.9E+16	5.0E+16	5.0E+16	100.9	101.1	1.2E+16	1.2E+16	24.8	24.5
2.5	5.2E+16	5.1E+16	5.2E+16	99.3	99.5	1.4E+16	1.4E+16	26.4	26.5
2.7	5.3E+16	5.4E+16	4.5E+16	100.6	83.8	1.5E+16	3.6E+16	28.8	67.2
2.9	5.4E+16	5.4E+16	5.4E+16	100.0	100.1	1.7E+16	1.7E+16	31.2	31.7
3.1	5.3E+16	5.3E+16	5.4E+16	100.0	101.1	1.7E+16	1.8E+16	32.7	33.3
3.3	5.3E+16	5.3E+16	5.2E+16	99.9	99.0	1.8E+16	1.8E+16	33.9	34.9
3.5	5.1E+16	5.1E+16	5.2E+16	99.9	100.6	1.8E+16	1.8E+16	34.1	34.2
3.7	5.0E+16	5.0E+16	5.0E+16	100.0	99.2	1.7E+16	1.7E+16	34.6	34.4
3.9	4.8E+16	4.8E+16	4.9E+16	100.0	100.9	1.7E+16	1.7E+16	35.0	34.4
4.1	4.7E+16	4.7E+16	4.6E+16	100.1	99.4	1.7E+16	1.6E+16	35.5	35.2
4.3	4.5E+16	4.5E+16	3.6E+16	99.9	81.1	1.6E+16	5.3E+16	36.1	117.9
4.5	4.3E+16	4.3E+16	4.2E+16	99.7	98.2	1.6E+16	1.6E+16	36.5	37.5
4.7	4.1E+16	4.2E+16	4.2E+16	100.5	101.2	1.5E+16	1.5E+16	36.6	36.1
4.9	4.0E+16	4.0E+16	3.9E+16	99.7	99.4	1.4E+16	1.4E+16	35.3	35.0

The results for test 4.5.3.3 are obtained by performing 1000 iterations of the vector reconstruction algorithm using a noiseless detector image of the hot-spots fireball. The results for test 4.5.3.4 are are obtained by performing 1000 iterations of the vector atmospheric reconstruction algorithm with a detector image that is atmospherically attenuated. The original and reconstructed input data cubes for both test cases are shown in Figure 5.38.

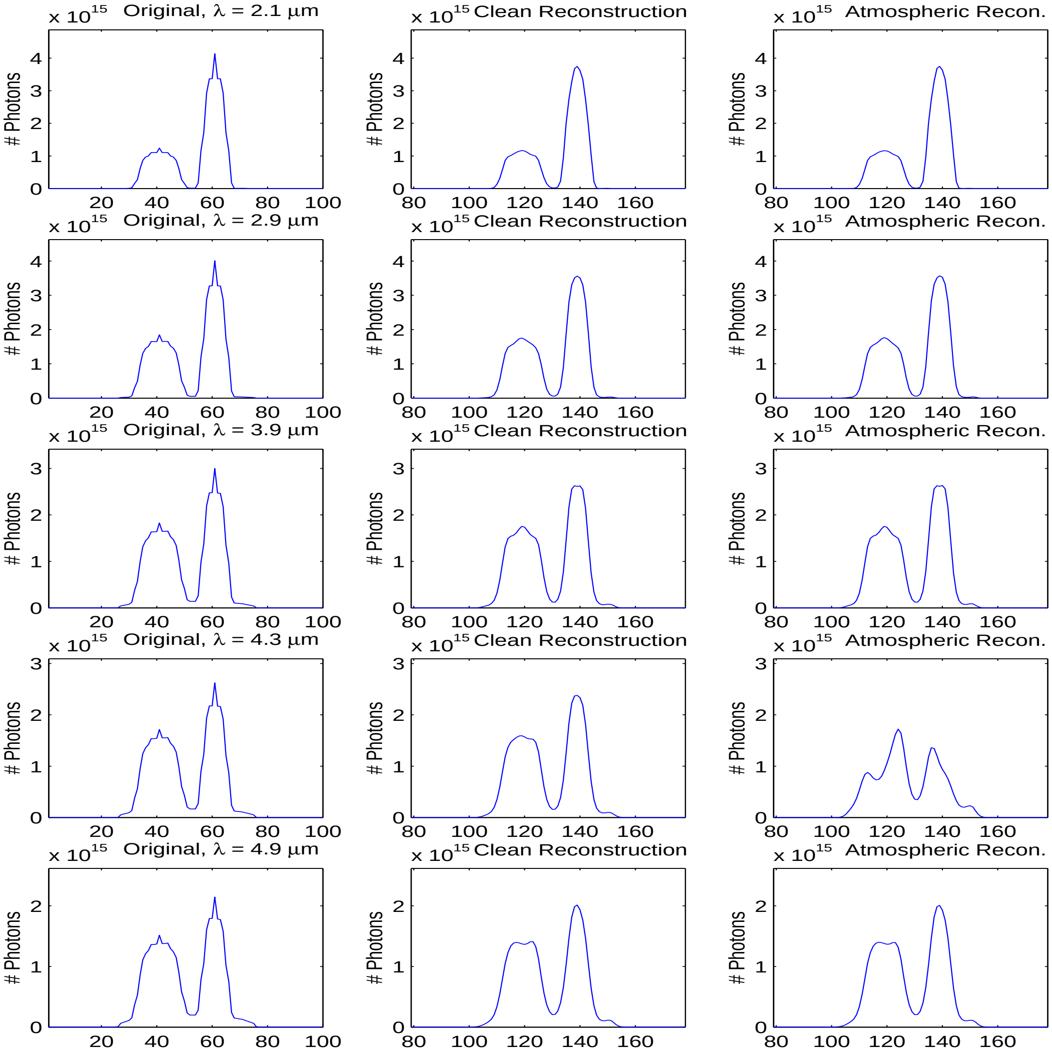


Figure 5.38: This figure shows the reproduction of the original hot-spots fireball data cube by the vector reconstruction algorithm. The reconstruction of atmospherically attenuated input data is also shown with degraded reconstruction performance in the $\lambda = 4.3\mu m$ bin.

The photon sums in each bin are shown in Figure 5.39 and demonstrate good tracking of the vector reconstruction algorithms on the clean input data. Reconstruction of the photon sums is also good for the atmospherically attenuated input data but degrades for the low valued $t_{atm}(\lambda)$ bins at $\lambda = 2.7\mu m$ and $\lambda = 4.3\mu m$.

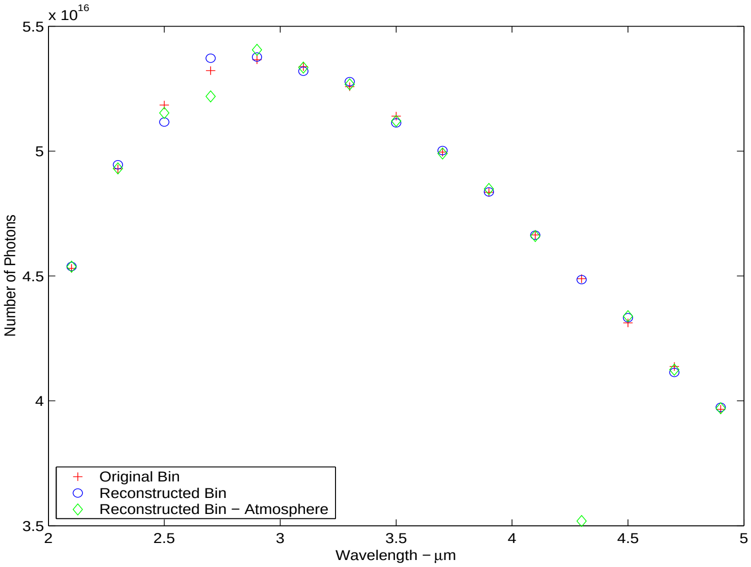


Figure 5.39: This figure shows the sum of photons in each wavelength bin for the hot-spots fireball in the reconstructed data compared with the original data cube. The sum of photons for the reconstruction of atmospherically attenuated detector data is also shown.

The remaining reconstruction results are shown in Table 5.31. The data indicates little drop-off in reconstruction performance across most wavelength bins when the atmospheric attenuation is included, with the exception of the $\lambda = 4.3\mu m$ bin.

A summary of the results for the concentric temperature rings fireball test cases 4.5.3.1 to 4.5.3.4 is presented in Table 5.32. The table indicates only a slight performance degradation when atmospherically attenuated detector data is used in both the 2D and vector reconstruction algorithms. The reconstruction metrics for the low valued $t_{atm}(\lambda)$ bins are also shown.

The temperatures of each spot and the background are also estimated from the reconstructed photon data for each of the 2D test cases as shown in Table 5.33. The estimated temperature is accurate to within 5% for each of the reconstructed spots and the background.

Table 5.31: Hot-Spots Fireball Vector Reconstruction Results For Clean And Atmospherically Attenuated Data

Bin λ	**Orig** **Bin**	**Recon** **Bin**	**Recon** **Bin A**	**R/O** **%**	**R/O** **% A**	**Error** **Photons**	**Error** **Photons A**	**Error** **%**	**Error** **% A**
2.1	4.5E+16	4.5E+16	4.5E+16	100.2	100.2	3.1E+15	3.0E+15	6.7	6.7
2.3	4.9E+16	4.9E+16	4.9E+16	100.3	100.0	3.5E+15	3.5E+15	7.0	7.0
2.5	5.2E+16	5.1E+16	5.2E+16	98.7	99.4	3.5E+15	3.4E+15	6.7	6.6
2.7	5.3E+16	5.4E+16	5.2E+16	100.9	98.1	3.4E+15	6.3E+15	6.4	11.9
2.9	5.4E+16	5.4E+16	5.4E+16	100.2	100.7	3.3E+15	3.3E+15	6.1	6.2
3.1	5.3E+16	5.3E+16	5.3E+16	99.7	99.9	3.2E+15	3.2E+15	6.0	5.9
3.3	5.3E+16	5.3E+16	5.3E+16	100.4	100.2	3.2E+15	3.1E+15	6.1	6.0
3.5	5.1E+16	5.1E+16	5.1E+16	99.5	99.6	3.0E+15	3.0E+15	5.9	5.9
3.7	5.0E+16	5.0E+16	5.0E+16	100.1	99.9	3.0E+15	2.9E+15	5.9	5.9
3.9	4.8E+16	4.8E+16	4.8E+16	100.0	100.3	2.8E+15	2.8E+15	5.9	5.8
4.1	4.7E+16	4.7E+16	4.7E+16	100.0	99.9	2.8E+15	2.8E+15	5.9	5.9
4.3	4.5E+16	4.5E+16	3.5E+16	99.9	78.4	2.7E+15	1.9E+16	6.0	42.9
4.5	4.3E+16	4.3E+16	4.3E+16	100.5	100.6	2.9E+15	3.1E+15	6.6	7.2
4.7	4.1E+16	4.1E+16	4.1E+16	99.4	99.7	2.9E+15	2.7E+15	6.9	6.4
4.9	4.0E+16	4.0E+16	4.0E+16	100.2	100.1	2.6E+15	2.5E+15	6.6	6.4

Table 5.32: Hot-Spots Fireball Results Summary

Test Case	**Photon Sum Error** **%**	**REM** **%**
2D	0.0 - 0.7	22.1 - 36.6
2D + atm	0.1 - 1.2	21.7 - 37.5
2D + atm: bad bins	16.2, 18.9	67.2, 117.9
1D	0.0 - 1.3	5.9 - 7.0
1D + atm	0.0 - 0.7	5.8 - 7.2
1D + atm: bad bins	1.9, 21.6	11.9, 42.9

Table 5.33: Hot-Spots Fireball Reconstructed Temperatures from 2D Reconstruction Algorithm

	Reconstructed Fireball Temperature			
Original **Temperature Ring** **K**	**Clean Data** **Estimate** **K**	**Error** **%**	**Atmospheric Data** **Estimate** **K**	**Error** **%**
1600	1665	4.1	1657	3.6
1500	1476	-1.6	1475	-1.7
1200	1176	-2.0	1179	-1.8
1000	994	-0.6	994	-0.6
600	571	-4.8	609	1.5
300	299	-0.3	292	-2.7

The results for test 4.6.1 are obtained by performing 1000 iterations of the vector reconstruction algorithm using a noiseless detector image of the first evolving fireball. The original and reconstructed input data cubes for test 4.6.1 are shown in Figure 5.40. Note that the original data consists of an upper and lower vector corresponding to the fireball at 1600^{o}K and 1500^{o}K respectively.

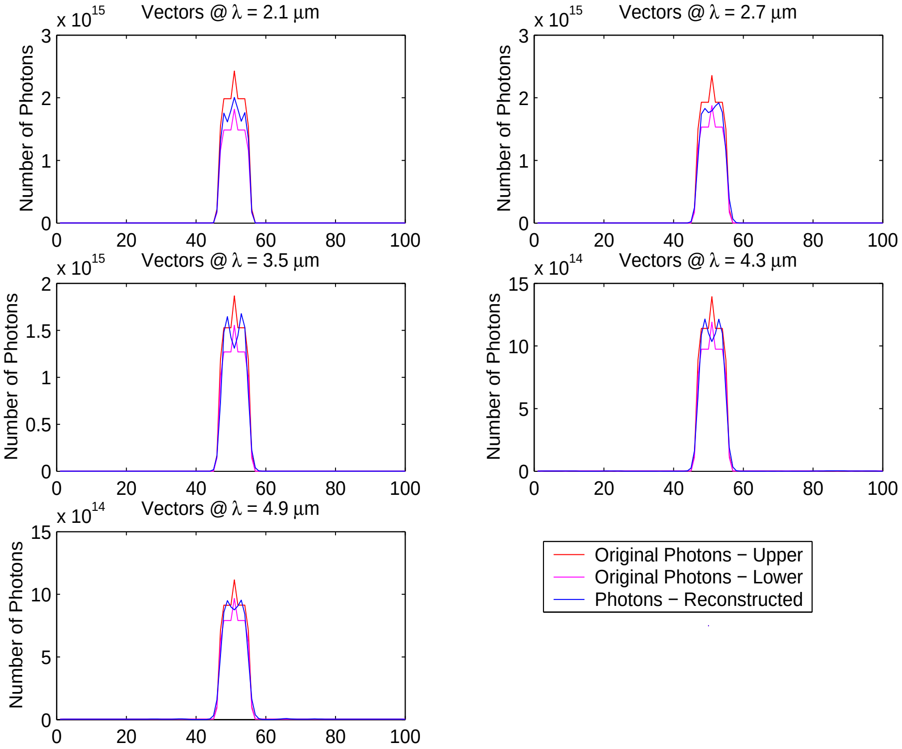


Figure 5.40: This figure shows the reproduction of the first evolving fireball data cube by the vector reconstruction algorithm. The reconstructed vector forms between the upper and lower photon levels and hence is averaging the vector of the original fireball.

The photon sums in each bin are shown in Figure 5.41 and demonstrates the reconstruction averaging the photons levels in the two original component fireball data cubes. The reconstruction results in Table 5.34 also show the averaging effect of the vector reconstruction algorithm when acting on the temporally evolving input data.

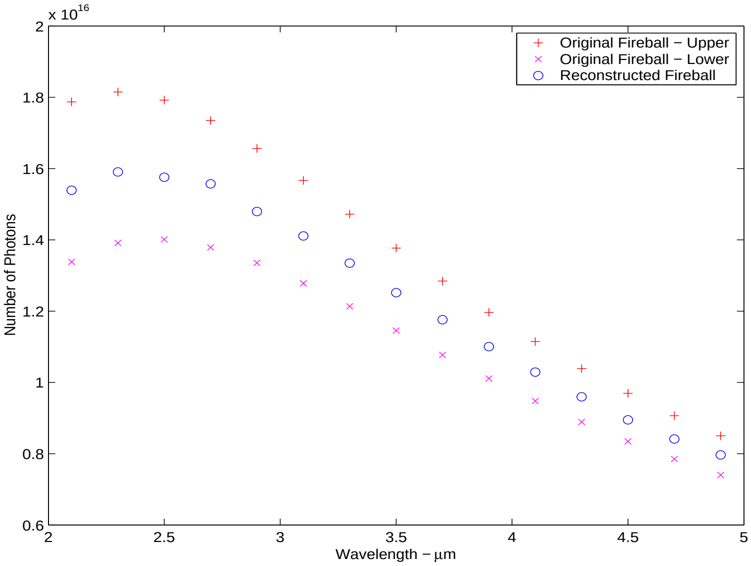


Figure 5.41: This figure shows the sum of photons in each wavelength bin for the first evolving fireball. The reconstruction algorithm averages the two components of the original input data cube.

Table 5.34: Evolving Fireball 1 Vector Reconstruction Results

Bin λ	**Orig** **Bin - Upper**	**Orig** **Bin - Lower**	**Recon** **Bin**	**R/O$_{\text{AV}}$** **%**	**Error$_{\text{AV}}$** **Photons**	**Error$_{\text{AV}}$** **%**
2.1	1.8E+16	1.3E+16	1.5E+16	98.5	6.2E+14	4.0
2.3	1.8E+16	1.4E+16	1.6E+16	99.2	8.7E+14	5.4
2.5	1.8E+16	1.4E+16	1.6E+16	98.7	1.1E+15	7.2
2.7	1.7E+16	1.4E+16	1.6E+16	100.0	1.7E+15	11.0
2.9	1.7E+16	1.3E+16	1.5E+16	98.9	1.9E+15	13.0
3.1	1.6E+16	1.3E+16	1.4E+16	99.2	2.1E+15	15.1
3.3	1.5E+16	1.2E+16	1.3E+16	99.4	2.1E+15	15.6
3.5	1.4E+16	1.1E+16	1.3E+16	99.3	2.0E+15	15.7
3.7	1.3E+16	1.1E+16	1.2E+16	99.6	1.9E+15	15.9
3.9	1.2E+16	1.0E+16	1.1E+16	99.7	1.7E+15	15.5
4.1	1.1E+16	9.5E+15	1.0E+16	99.8	1.5E+15	14.8
4.3	1.0E+16	8.9E+15	9.6E+15	99.6	1.4E+15	14.4
4.5	9.7E+15	8.3E+15	9.0E+15	99.2	1.2E+15	13.9
4.7	9.1E+15	7.9E+15	8.4E+15	99.4	1.2E+15	14.6
4.9	8.5E+15	7.4E+15	8.0E+15	100.2	1.1E+15	13.6

The results for test 4.6.2 are obtained by performing 1000 iterations of the vector reconstruction algorithm using a noiseless detector image of the first evolving fireball. The original and reconstructed input data cubes for test 4.6.2 are shown in Figure 5.42. Note that the original data consists of an upper and lower vector corresponding to the fireball at 1600^{o}K and 1300^{o}K respectively.

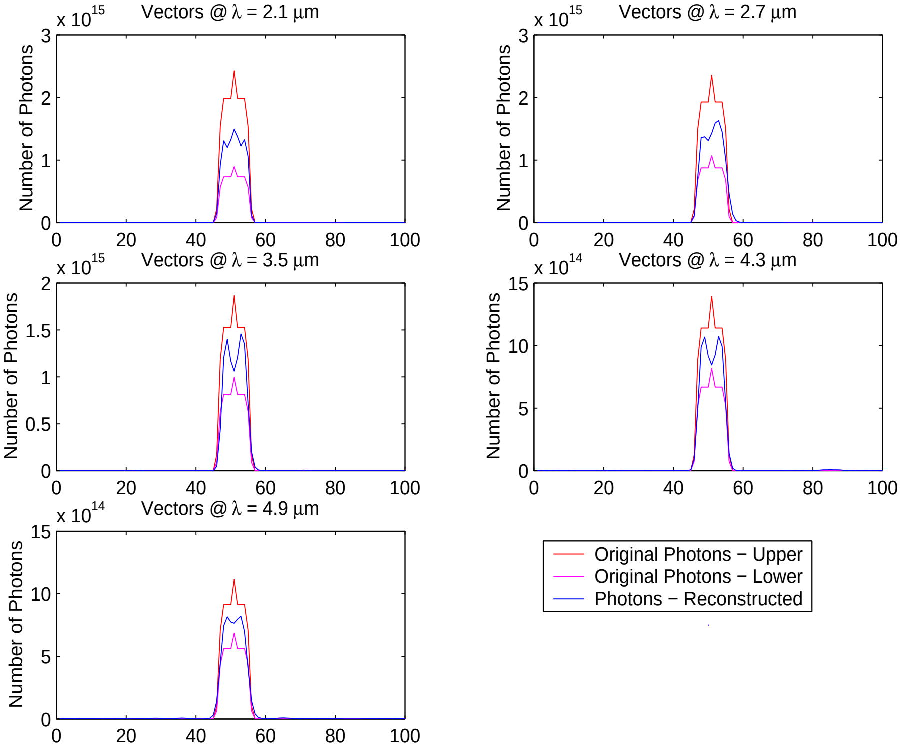


Figure 5.42: This figure shows the reproduction of the second evolving fireball data cube by the vector reconstruction algorithm. The reconstructed vector forms between the upper and lower photon levels and hence is averaging the vector of the original fireball.

The photon sums in each bin are shown in Figure 5.43 and demonstrates the reconstruction averaging the photons levels in the two original component fireball data cubes. The reconstruction results in Table 5.35 also show a slight degradation compared to the previous test case attributed to the larger temperature difference between the fireball components.

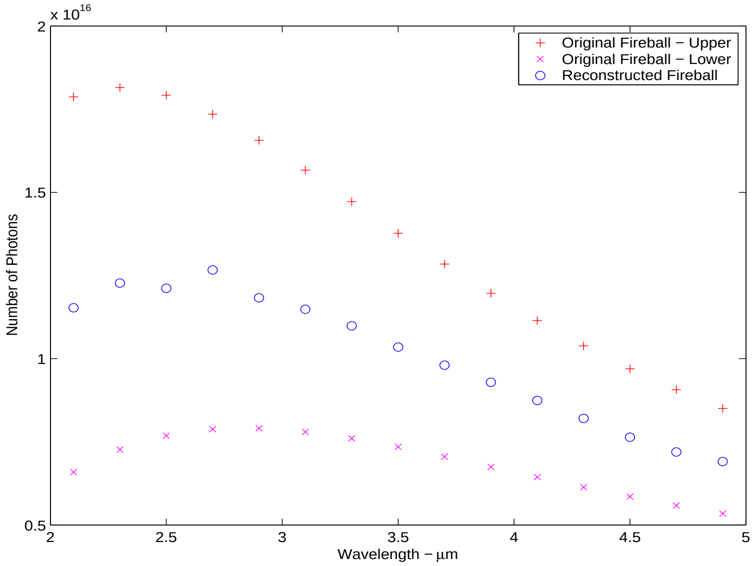


Figure 5.43: This figure shows the sum of photons in each wavelength bin for the second evolving fireball. The reconstruction algorithm averages the two components of the original input data cube.

Table 5.35: Evolving Fireball 2 Vector Reconstruction Results

Bin	**Orig**	**Orig**	**Recon**	**R/O$_{\mathbf{AV}}$**	**Error$_{\mathbf{AV}}$**	**Error$_{\mathbf{AV}}$**
λ	**Bin - Upper**	**Bin - Lower**	**Bin**	**%**	**Photons**	**%**
2.1	1.8E+16	6.6E+15	1.2E+16	94.3	7.7E+14	6.3
2.3	1.8E+16	7.3E+15	1.2E+16	96.6	9.6E+14	7.6
2.5	1.8E+16	7.7E+15	1.2E+16	94.6	1.5E+15	11.8
2.7	1.7E+16	7.9E+15	1.3E+16	100.4	1.9E+15	15.2
2.9	1.7E+16	7.9E+15	1.2E+16	96.7	2.0E+15	16.5
3.1	1.6E+16	7.8E+15	1.1E+16	97.9	2.4E+15	20.2
3.3	1.5E+16	7.6E+15	1.1E+16	98.4	2.0E+15	18.0
3.5	1.4E+16	7.3E+15	1.0E+16	98.0	2.0E+15	19.3
3.7	1.3E+16	7.1E+15	9.8E+15	98.6	1.9E+15	19.5
3.9	1.2E+16	6.7E+15	9.3E+15	99.3	1.7E+15	18.4
4.1	1.1E+16	6.4E+15	8.7E+15	99.4	1.6E+15	17.9
4.3	1.0E+16	6.1E+15	8.2E+15	99.3	1.4E+15	16.4
4.5	9.7E+15	5.9E+15	7.6E+15	98.3	1.2E+15	15.6
4.7	9.1E+15	5.6E+15	7.2E+15	98.2	1.1E+15	14.5
4.9	8.5E+15	5.3E+15	6.9E+15	99.8	1.0E+15	14.8

The results for test 4.6.3 are obtained by performing 1000 iterations of the vector reconstruction algorithm using a noiseless detector image of the first evolving fireball. The original and reconstructed input data cubes for test 4.6.3 are shown in Figure 5.44. Note that the original data consists of an upper, mid and lower vector corresponding to the fireball at 1600^{o}K, 1400^{o}K and 1200^{o}K respectively.

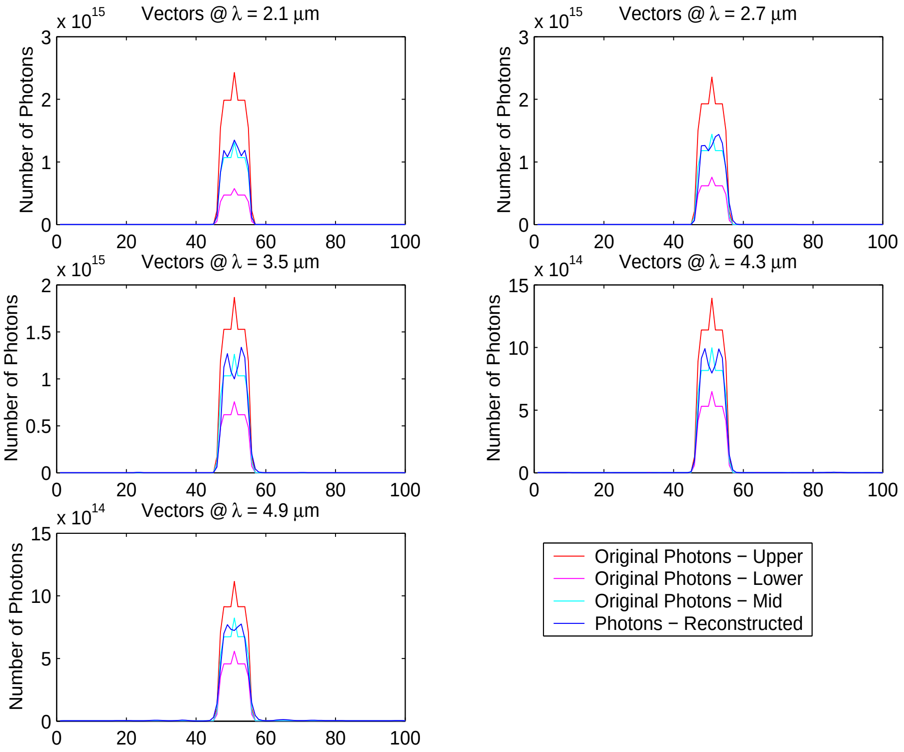


Figure 5.44: This figure shows the reproduction of the third evolving fireball data cube by the vector reconstruction algorithm. The reconstructed vector forms near the middle photon level which is the center and also average of the original fireball.

The photon sums in each bin are shown in Figure 5.45 and demonstrates the reconstruction averaging the photons levels in the three original component fireball data cubes. The reconstruction results are also shown in Table 5.36.

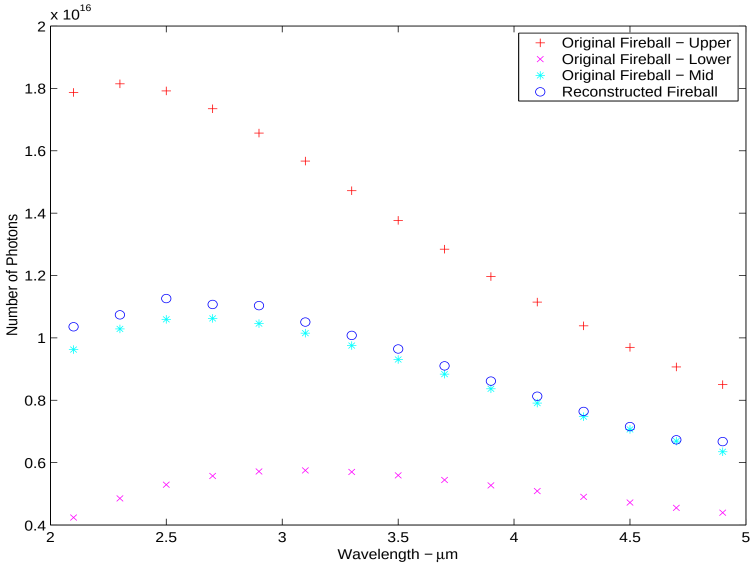


Figure 5.45: This figure shows the sum of photons in each wavelength bin for the third evolving fireball. The reconstruction algorithm is averages the three components of the original input data cube.

Table 5.36: Evolving Fireball 3 Vector Reconstruction Results

Bin	**Orig**	**Orig**	**Recon**	**R/O$_{AV}$**	**Error$_{AV}$**	**Error$_{AV}$**
λ	**Bin - Upper**	**Bin - Lower**	**Bin**	**%**	**Photons**	**%**
2.1	1.8E+16	4.2E+15	1.0E+16	97.9	5.0E+14	4.7
2.3	1.8E+16	4.9E+15	1.1E+16	96.8	8.1E+14	7.3
2.5	1.8E+16	5.3E+15	1.1E+16	99.9	1.1E+15	9.7
2.7	1.7E+16	5.6E+15	1.1E+16	99.0	1.6E+15	14.1
2.9	1.7E+16	5.7E+15	1.1E+16	101.1	1.7E+15	15.8
3.1	1.6E+16	5.8E+15	1.1E+16	99.8	2.0E+15	19.4
3.3	1.5E+16	5.7E+15	1.0E+16	100.2	1.9E+15	18.4
3.5	1.4E+16	5.6E+15	9.6E+15	100.9	1.8E+15	19.2
3.7	1.3E+16	5.4E+15	9.1E+15	100.6	1.7E+15	19.1
3.9	1.2E+16	5.3E+15	8.6E+15	100.9	1.6E+15	18.4
4.1	1.1E+16	5.1E+15	8.1E+15	101.0	1.4E+15	17.7
4.3	1.0E+16	4.9E+15	7.6E+15	100.7	1.3E+15	16.6
4.5	9.7E+15	4.7E+15	7.2E+15	99.9	1.1E+15	16.0
4.7	9.1E+15	4.6E+15	6.7E+15	99.5	1.0E+15	15.5
4.9	8.5E+15	4.4E+15	6.7E+15	104.0	1.0E+15	16.3

The results for test 4.6.4 are obtained by performing 1000 iterations of the vector reconstruction algorithm using a noiseless detector image of the first evolving fireball. The original and reconstructed input data cubes for test 4.6.4 are shown in Figure 5.46. Note that the original data consists of an upper, mid and lower vector corresponding to the fireball at 1600^{o}K, 1000^{o}K and 400^{o}K respectively.

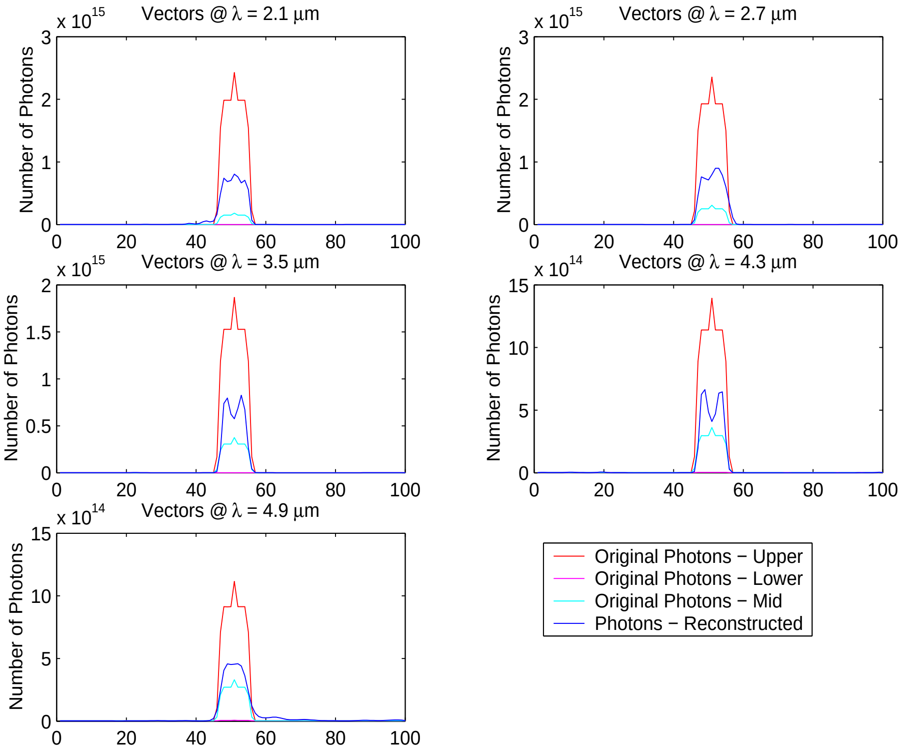


Figure 5.46: This figure shows the reproduction of the fourth evolving fireball data cube by the vector reconstruction algorithm. The reconstructed vector forms above the middle photon level as it is biased by the low photon levels for the 400^{o}K component data.

The photon sums in each bin are shown in Figure 5.47 and show a degradation compared to the previous test case. This is attributed to the large temperature differences between the original fireball components resulting in different Planckian profiles of fireball. The reconstruction results are also shown in Table 5.37.

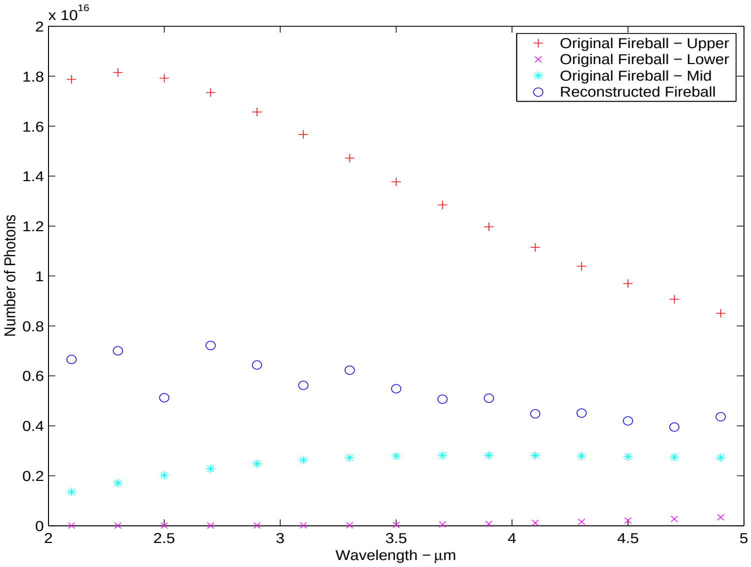


Figure 5.47: This figure shows the sum of photons in each wavelength bin for the fourth evolving fireball. The reconstruction algorithm is averages the three components of the original input data cube with a greater error than in the previous test case.

Table 5.37: Evolving Fireball 4 Vector Reconstruction Results

Bin λ	**Orig** **Bin - Upper**	**Orig** **Bin - Lower**	**Recon** **Bin**	**R/O**$_{\mathbf{AV}}$ **%**	**Error**$_{\mathbf{AV}}$ **Photons**	**Error**$_{\mathbf{AV}}$ **%**
2.1	1.8E+16	7.6E+10	6.7E+15	103.9	6.9E+14	10.8
2.3	1.8E+16	2.7E+11	7.0E+15	105.8	5.0E+14	7.5
2.5	1.8E+16	8.2E+11	5.1E+15	77.1	1.6E+15	24.0
2.7	1.7E+16	2.1E+12	7.2E+15	110.3	1.1E+15	16.9
2.9	1.7E+16	4.8E+12	6.4E+15	101.3	6.2E+14	9.8
3.1	1.6E+16	9.6E+12	5.6E+15	92.2	1.2E+15	19.0
3.3	1.5E+16	1.8E+13	6.2E+15	107.0	1.3E+15	23.2
3.5	1.4E+16	3.1E+13	5.5E+15	99.2	1.4E+15	25.3
3.7	1.3E+16	5.0E+13	5.1E+15	96.7	1.6E+15	29.9
3.9	1.2E+16	7.5E+13	5.1E+15	103.0	1.6E+15	31.9
4.1	1.1E+16	1.1E+14	4.5E+15	95.6	1.6E+15	33.2
4.3	1.0E+16	1.5E+14	4.5E+15	101.5	1.3E+15	29.6
4.5	9.7E+15	2.0E+14	4.2E+15	99.4	1.1E+15	27.1
4.7	9.1E+15	2.7E+14	3.9E+15	98.1	1.2E+15	29.8
4.9	8.5E+15	3.4E+14	4.4E+15	113.1	1.0E+15	26.5

A summary of the results for the evolving fireball test cases 4.6.1 to 4.6.4 is presented in Table 5.38. The reconstruction metrics are given relative to the average of the input hyperspectral data cube. The table also shows the reconstructed temperature which again demonstrates the vector reconstruction algorithm producing the average of the input data. As discussed, the performance drop-off in the final evolving fireball is attributed to the large temperature differences between the original fireball components.

Table 5.38: Evolving Fireball Results Summary and Reconstructed Temperatures

Test Case	Photon Sum Error %	REM %	Reconstructed Temperature K	Temperature Error %
1600K to 1500K	0.0 - 1.5	4.0 - 15.9	1515	-2.3
1600K to 1300K	0.4 - 3.4	6.3 - 20.2	1436	-1.0
1600K to 1400K to 1200K	0.2 - 4.0	4.7 - 19.4	1402	0.1
1600K to 1000K to 400K	0.6 - 22.9	7.5 - 33.2	1233	23.3

5.8 Analysis of Two-Dimensional Reconstruction Results

A summary of the 2D reconstruction results in this chapter is presented as follows:

- Spatial features of the original input data cube are reconstructed with blurring induced by the CTIS optics.
- The Reconstruction Error Metric (REM) is more pronounced for point sources (binary stars: 2D: 16.5 to 148%) compared to extended sources (overlapping bars: 2D: 4.2 to 18.7%). This trend increases as the wavelength increases, ie, it is directly proportional to the spatial extent of the PSF which characterizes the optical blurring.
- Reconstruction of the total number of photons in each spectral bin is achievable enabling absolute radiometry calculations in both blackbody point and extended sources. The absolute radiometry is used to reconstruct the temperature of features in the imaged scene to within an error margin of 3.9% for point sources (binary stars) and 4.1% for extended sources (concentric rings fireball, hot-spots fireball).
- Reconstruction of the total number of photons is successful for atmospherically attenuated data where the atmospheric transmission coefficient is greater than 3%, ie, $t_{atm}(\lambda) > 0.03$. These low valued atmospheric transmission coefficient bins can be masked from the reconstructed temperature estimation resulting in reconstructed temperature estimates to within 3.6% of the nominal scene temperature (concentric rings fireball, hot-spots fireball).
- Reconstruction of the total number of photons in each wavelength bin on atmospherically attenuated detector data produces little degradation (binary stars: 0.0%, monochromatic numbers: 1.7%, concentric rings fireball: 2.9%, hot-spots fireball: 0.5%) in reconstruction performance compared with pristine detector data. This is applicable for the wavelength bins where the atmospheric transmission coefficient is greater than 3%, ie, $t_{atm}(\lambda) > 0.03$.

- There is little variation (binary star: 2.5%, monochromatic numbers: 1.3%, concentric rings fireball: 2.9%, hot-spots fireball: 0.9%) in the reconstruction of spatial features, as measured by the REM, when the atmospheric attenuation is included. This is again applicable for the wavelength bins where the atmospheric transmission coefficient is greater than 3%, ie, $t_{atm}(\lambda) > 0.03$.
- Reconstruction of the total number of photons with noisy detector data shows little drop-off in performance compared to the noiseless detector data (separate bars: 1%, overlapping bars: 5.2%, monochromatic numbers: 0.2%) .
- The REM slightly increases when reconstructing a noisy detector image compared to noiseless detector data (separate bars: 4.6%, overlapping bars: 6.5%, monochromatic numbers: 1.2%)
- The 2D reconstruction algorithm can spectrally separate spatially overlapping sources. With increasing spatial information and decreasing spatial correlation between bins, some spectral bleeding may occur between adjacent bins (separate bars: 2.2%, overlapping bars: 6.4%) .

5.9 Analysis of Vector Reconstruction Results

A summary of the vector reconstruction results in this chapter is presented as follows:

- Spatial features of the vector in the original data cube are reconstructed with additional diffractive blurring induced by the CTIS optics.
- The Reconstruction Error Metric (REM) is more pronounced for point sources (binary stars: 1D: 1.8 to 88.0%) compared to extended sources (overlapping bars: 1D: 4.2 to 48.2%).
- For the input data cube with spatial dimensions of 100 × 100 pixels, the REM increases directly proportionally to the level of spatial features in the data cube (uniform fireballs: 3.9% to 17.8%, separate bars: 5.9% to 29.3% , overlapping bars: 4.2 to 48.2%).

- Reconstruction of the total number of photons in an input data cube with spatial dimensions of 100 × 100 pixels can show a significant drop-off in performance when compared to the results of the equivalent 2D reconstruction (separate bars: 23.4%, overlapping bars: 33.1%).

- Reconstruction of the total number of photons in an input data cube with spatial dimension of 20×20 pixels is successful with highly comparable results to the equivalent 2D reconstruction (separate bars, overlapping bars).

- For clean detector data, the reconstructed temperature estimate is within 1.6% of the nominal temperature for blackbody point sources (binary star) and within 6.0% for extended blackbody sources (uniform fireballs).

- For an input data cube with spatial dimensions of 100 × 100 pixels, spectral bleeding significantly increases in the vector reconstruction compared the 2D reconstruction (separate bars: 10.0%, overlapping bars: 13.8%).

- Reconstruction of the total number of photons is successful for atmospherically attenuated data where the atmospheric transmission coefficient is greater than 3%, ie, $t_{atm}(\lambda) > 0.03$.

- Of the two methods investigated, inclusion of the atmospheric data within the reconstruction algorithm produces superior results in scene temperature reconstruction over the direct post-processing atmospheric inversion method (uniform fireball: 2.1% temperature error compared to 23.4%).

- For atmospherically attenuated detector data, the reconstructed temperature estimate is within 0.8% of the nominal temperature for blackbody point sources (binary star) and within 5.8% for extended blackbody sources (uniform fireballs). This accuracy is achieved when the low valued $t_{atm}(\lambda)$ are masked from the temperature estimate calculation.

- For a noisy detector image of an input data cube with spatial dimensions of 100×100 pixels, there is little drop-off in REM compared with a noiseless detector image in reconstructing the total number of photons in each wavelength bin (separate bars: 6.3%, overlapping bars: 3.1%).

- For a noisy detector image of an input data cube with spatial dimensions of 20×20 pixels, degradation in performance of reconstructing spatial features as indicated by the REM can be significant (separate bars: 44.1%, overlapping bars: 23.7%). This indicates a higher susceptibility to noise of the smaller input data cube compared to the 100×100 input data cube.

- Reconstruction of total photon numbers in each wavelength bin and the follow-on temperature estimates do not degrade with the use of a non-zero temperature background in the input temperature scene (binary star, uniform fireballs).

- The vector reconstruction algorithm can be applied to hyperspectral data from a temporally evolving scene. This data category implies that the incident photon flux on the DVP is not constant throughout one complete revolution of the DVP. For temporally evolving input data the vector reconstruction algorithm will estimate the temporal average of the input data cube.

- An estimate of the reconstructed temperature at a pixel location in a temporally evolving scene can be within 2.3% of the average temperature of the original scene location (evolving fireball).

VI. Conclusions and Recommendations

This chapter restates the research goals and provides a summary of the research conducted in this thesis. Conclusions are documented on the performance of Estimation Theory based reconstruction algorithms in estimating the hyperspectral content of an input scene. Significant research results and recommendations for further work are also provided.

6.1 Restatement of Research Goals

The research goals in section 1.3 of this thesis are restated as follows:

- The development and evaluation of a CTIS detector reconstruction algorithm which maintains absolute radiometric accuracy.
- The development and evaluation of a CTIS detector reconstruction algorithm capable of imaging a temporally evolving input image scene.
- The investigation of two-dimensional versus one-dimensional CTIS detector reconstruction algorithms in terms of absolute radiometric performance.
- The development and evaluation of a CTIS detector reconstruction algorithm which can be applied to atmospherically attenuated detector data.
- The evaluation of the CTIS detector reconstruction algorithms in the presence of photon noise at the detector.

6.2 Research Summary

Chapter I describes the concept of hyperspectral imagery and the format of the associated hyperspectral data cubes. Hyperspectral imagery has applications in diverse imaging scenarios providing substantial motivation to conduct further research in this area. The specific research goals of this thesis are stated in this chapter along with an overview of the organization of this document.

Chapter II discusses conventional hyperspectral collection techniques and their inherent disadvantages. The concept and development of the Chromotomographic Imaging System (CTIS) is presented. The CTIS optical configuration and post-processing requirements are documented along with its concept of operation for collecting hyperspectral data.

Chapter III develops the discrete model of the CTIS. It includes the modelling assumptions and design parameters used to implement the model. An optical propagation model of both the POS and DOS are developed to simulate CTIS-like detector images. The 2D and vector input hyperspectral reconstruction algorithms are derived along with methods for including the effects of atmospheric attenuation on the detector images.

Chapter IV establishes the test cases used to verify the performance of the reconstruction algorithms. For each test case an input hyperspectral data cube for the imaging scenario is generated. The effect of atmospheric attenuation is also applied to several of the input data cubes. The input object data cubes, $O(\lambda, u, v)$, are used as the input to the CTIS model which produces the corresponding CTIS detector images. Additional photon noise is also applied to several of the detector images. The detector images provide the input to the reconstruction algorithms which provide an estimate, $\widehat{O}(\lambda, u, v)$, of the original input hyperspectral data cube.

Chapter V demonstrates the performance of the reconstruction algorithms in estimating the input hyperspectral data cube for each test case in chapter IV. Reconstruction is performed on both the 2D detector image and the 1D x vector sum for several of the test cases. The treatment of atmospheric attenuation and the addition of noisy detector data is also demonstrated in the test cases. The chapter concludes with an analysis of the performance of the 2D and vector reconstruction algorithms.

6.3 Conclusions

6.3.1 2D Reconstruction Performance. The results in chapter V show good performance of the 2D reconstruction algorithm for a variety of test cases. Temperature estimates calculated from the reconstructed hyperspectral data cube are within 3.9% of the original temperature for blackbody point sources and within 4.1% for extended blackbody sources. The algorithm also spectrally separates spatially overlapping monochromatic extended sources.

6.3.2 Vector Reconstruction Performance. The vector reconstruction algorithm is viewed as a method for recording and post-processing hyperspectral data at faster rate than required by the 2D reconstruction algorithm. Though spatial features in the reconstruction are reduced to 1D, there may be applications where this is sufficient. Temperature estimates calculated from the reconstructed hyperspectral vector data are within 1.6% of the original temperature for blackbody point sources and within 6.0% for extended blackbody sources.

6.3.3 Reconstruction of Atmospherically Attenuated Detector Data. Both the 2D and vector reconstruction algorithm can be modified to account for atmospherically attenuated detector data. Absolute radiometric accuracy is maintained in the reconstructed data cube in the wavelength bins where the atmospheric transmission coefficient is greater than 3%. Radiometric accuracy improves by an order of magnitude when the atmospheric attenuation data is included within the reconstruction algorithm. This is opposed to directly inverting the atmospheric attenuation of the processed data cube following completion of the reconstruction iterations. Temperature estimates calculated from the atmospherically attenuated detector data are within 3.6% of the original temperature for 2D reconstructions and within 5.8% for vector reconstructions.

6.3.4 Reconstruction of Noisy Detector Data. Both the 2D and vector reconstruction algorithm perform well in the presence of noisy detector data showing little performance degradation compared with noiseless detector data. The configuration of imaging a input hyperspectral data cube with a smaller spatial extent is more susceptible to detector noise.

6.3.5 Reconstruction of Temporally Evolving Detector Data. The vector reconstruction algorithm without any modification will reconstruct the temporal average of a temporally evolving input hyperspectral data cube. The reconstructed temperature at a pixel location in a temporally evolving scene can be within 2.3% of the average temperature of the original scene location. When paired with the high optical throughput of the CTIS, the faster collection of scene vector sum data and the faster processing time of the vector reconstruction algorithm, it is conceivable to apply a CTIS instrument in a real-time hyperspectral imaging application. This application could utilize a vector reconstruction as a first-look scan of a scene and then capture 2D hyperspectral data to increase the spatial information of points of interest within the scene.

6.4 Significant Results of Research

6.4.1 Viability of an Estimation Theory Based CTIS Reconstruction Algorithm. This research investigates an alternate approach to deterministic matrix inversion techniques in reconstructing hyperspectral data cubes from a CTIS detector image. The results in chapter V show the good performance of an Estimation Theory based reconstruction algorithm for reconstructing the absolute radiometry and spatial features of a hyperspectral input data cube. These algorithms also do not significantly degrade in the presence of noisy detector data. The radiometrically accurate reconstruction of atmospherically attenuated detector data proves viable for wavelengths where there is sufficient photon levels at the detector. The algorithm also exhibits stable performance behaviour when reconstructing a temporally evolving hyperspectral input data cube.

6.4.2 Trade-off Between 2D and Vector Reconstruction Algorithms. This research investigates comparable reconstructions of 2D and vector sum detector data. The 2D reconstructions maintain absolute radiometric accuracy and reproduce the spatial features of the scene input hyperspectral data cube. The vector reconstructions maintain absolute radiometric accuracy, but show some performance degradation as the spatial information in the scene increases. The spatial information reproduced by the vector reconstruction algorithm is limited to a 1D sum of the input scene. However the collection and post-processing of vector detector data can be performed at a much higher rate than the 2D reconstruction. This potentially allows for CTIS applications in real-time roles including the imaging temporally evolving scenes.

6.5 Recommendations for Future Work

6.5.1 Development of a CTIS Instrument. This research in developing radiometrically accurate CTIS reconstruction algorithms complements previous AFIT CTIS work. In conjunction with the propagation models developed by Dearinger in [6] and the sensor characterization work completed by LeMaster in [13], the design tools are established to develop an imaging application specific CTIS instrument. The overall design can be modelled with trade-off analyses undertaken between the optical component requirements and the reconstruction performance.

6.5.2 Optimization of the Reconstruction Algorithm Implementation. The reconstruction algorithms in this thesis are implemented in Matlab®. This implementation successfully demonstrates the processing steps required for the reconstruction. However the utility of particularly the 2D reconstruction algorithm could be vastly improved by increasing the processing speed of the algorithm. The advantages of faster processing speed include the manageable extension to more wavelength bins in the reconstruction and also the use of greater algorithm iteration numbers to improve the reconstruction performance.

Other optimization related potential algorithm modifications include investigating the effect of having the number of DVP rotation angles not equal to the number of wavelength bins. The inclusion of an algorithm stopping criteria could also be investigated. This would be designed to stop the algorithm iterating when the reconstructed hyperspectral data cube remains constant within some threshold between consecutive algorithm iterations.

6.5.3 Development of Additional Reconstruction Algorithms. There are potential extensions, or tailoring, to the reconstruction algorithms that could be investigated depending on potential specific imaging applications. One possible modification is to measure two orthogonal vector sums, ie, both the x and y directions, of the detector data and attempt to estimate the 2D input hyperspectral data cube. Another possible extension applicable to imaging blackbody sources is to directly estimate the temperature distribution in the scene using the Planck function within the estimation algorithm. Additional transfer functions describing optical aberrations induced by the CTIS optics and atmospheric turbulence could also be included within the estimation algorithm.

6.5.4 Modelled PSF Accuracy Sensitivity Study. The reconstruction algorithms used in this thesis assume perfect knowledge of the CTIS PSF. In a CTIS instrument there will be aberrations resulting from the optical components. Atmospheric turbulence will also increase the uncertainty in the overall scene and CTIS PSF. A study to investigate the effect of abberations on reconstruction performance could be implemented.

6.5.5 Study of Reconstruction Accuracy versus Number of Algorithm Iterations. The 2D reconstructions in this thesis all used 100 algorithm iterations while the vector reconstructions used 1000 iterations. This is to provide a consistent simulation set-up for each test case. A study to investigate and quantify the reconstruction accuracy improvement per iteration of the algorithm could be conducted.

Appendix A. Spectral Refractive Indices of Optical Materials

The spectral refractive indices for both LiF and BaFl_2 are published by Wolfe and Zissis in [22]. The refractive indices for the wavelength range modelled in this thesis are repeated as follows:

Table A.1: Spectral Refractive Indices of Lithium Fluoride (LiF)

Wavelength (λ) in μm	**Refractive Index**
2.0	1.37875
2.5	1.37327
3.0	1.36660
3.5	1.35868
4.0	1.34942
4.5	1.33875
5.0	1.32661

Table A.2: Spectral Refractive Indices of Barium Fluoride (BaF_2)

Wavelength (λ) in μm	**Refractive Index**
1.97009	1.46470
2.1526	1.46412
2.32542	1.46356
2.5766	1.46271
2.6738	1.46237
3.2434	1.46017
3.422	1.45941
5.138	1.45014

Appendix B. Additional Derivations

This appendix contains additional derivations of the mathematical models describing the physical phenomena simulated in this thesis.

B.1 Rayleigh-Sommerfeld Diffraction Sum for Wave Optics Propagation Through a Lens

As demonstrated by Cain in [4] the following derivation applies the continuous Rayleigh-Sommerfeld diffraction formula to the discrete domain of wave optics simulation. This derivation is specific to simulating a wave optics propagation from a lens to a detector array.

The Rayleigh-Sommerfeld diffraction integral, as presented by Goodman in chapter 3 of [10], describes the wave optics propagation of light between two parallel planes. This is mathematically expressed as

$$U(x_1, y_1) = \frac{1}{j\lambda} \iint\limits_{\Sigma} \frac{U(x_0, y_0)}{r_{01}} e^{jkr_{01}} \cos\theta \; dx_0 dy_0 \tag{B.1}$$

where $U(x_0, y_0)$ is the electromagnetic field over a transmitting aperture plane $\sum$ and $U(x_1, y_1)$ is the field at a receiving plane. The distance between points on the planes is denoted r_{01} and is a Pythagorean function of the perpendicular distance between the planes and transverse plane coordinates (x_0, y_0) and (x_1, y_1). The angle of r_{01} with respect to the propagation normal direction is denoted as θ. The integral is applicable to monochromatic light with wavelength λ and corresponding wave number $k = \frac{2\pi}{\lambda}$.

Propagation through a lens can be treated as a phase transformation in the electromagnetic field between the field entering and exiting the lens. The phase transformation for a lens, $t_l(x, y)$, is shown by Goodman in chapter 5 of [10] to be

$$t_l(x, y) = e^{\frac{-j2\pi(x^2+y^2)}{2\lambda f}} \tag{B.2}$$

where (x, y) are the transverse coordinates of the lens (perpendicular to direction of propagation), f is the lens focal length and λ is the wavelength of the light.

Upon substituting equation B.2 into equation B.1, the continuous expression for wave optics propagation through a lens becomes

$$U(x_1, y_1) = \frac{1}{j\lambda} \iint\limits_{\Sigma} t_l(x_0, y_0) \frac{U(x_0, y_0)}{r_{01}} e^{jkr_{01}} \cos\theta \; dx_0 dy_0 \tag{B.3}$$

Several assumptions are now required to proceed with casting the continuous integral of equation B.3 into a discrete sum that can be implemented as numerical simulation wave optics code. The Pythagorean distance r_{01} between the points in the transmitting, or lens, plane and the receiving, or detector plane, is expressed as

$$r_{01} = \sqrt{f^2 + (x - x_d)^2 + (y - y_d)^2} \tag{B.4}$$

where (x, y) are the continuous transverse coordinates in the lens plane, (x_d, y_d) are transverse coordinates at the detector and f is the lens focal length. Detectors are physically composed of an array of pixels which combine to collect the overall image. Hence (x_d, y_d) can be regarded as sampling coordinates of the field at the detector plane.

A paraxial approximation is now applied to r_{01} be rewriting it as

$$r_{01} = f\sqrt{1 + \frac{(x - x_d)^2 + (y - y_d)^2}{f^2}} \tag{B.5}$$

and using the Taylor series expansion of $\sqrt{1 + b} \approx 1 + \frac{b}{2} - \frac{b^2}{8} \;\cdots$, which results in

$$r_{01} \approx f\left(1 + \frac{(x - x_d)^2 + (y - y_d)^2}{2f^2}\right) \tag{B.6}$$

when the first two terms of the Taylor series are applied.

Also observe from trigonometry that

$$\cos\theta = \frac{f}{\sqrt{f^2 + (x - x_d)^2 + (y - y_d)^2}} \tag{B.7}$$

The continuous field at the lens must also be sampled in order to model the propagation. As such (x_l, y_l) are treated as discrete samples of the lens such that the continuous field $U(x, y)$ exiting the lens is represented as by the sampled field $U(x_l, y_l)$.

The approach for determining the discrete field at the detector begins by considering the contribution from a single sample of the lens. The field at the detector resulting from a single lens sample is denoted $U_{LS}(x_d, y_d)$.

The limits of the integration at equation B.3 correspond to the spatial extent of the lens sample being propagated. This can be expressed as $x_l \pm \frac{\Delta}{2}$ and $y_l \pm \frac{\Delta}{2}$ where Δ is the side-length of the lens sample with the assumption that the lens sample is square. This indexing places the lens sample coordinate (x_l, y_l) at the midpoint of the lens sample.

Using the paraxial approximation at equation B.6 in the exponential, the lens transformation at equation B.2 and substituting equation B.7 for $cos\theta$, equation B.3 represents $U_{LS}(x_d, y_d)$ as

$$\begin{aligned}
U_{LS}(x_d, y_d) \quad &= \quad \frac{1}{j\lambda} \int_{x_l - \frac{\Delta}{2}}^{x_l + \frac{\Delta}{2}} \int_{y_l - \frac{\Delta}{2}}^{y_l + \frac{\Delta}{2}} \frac{U(x_l, y_l)}{\sqrt{f^2 + (x - x_d)^2 + (y - y_d)^2}} \frac{f}{\sqrt{f^2 + (x - x_d)^2 + (y - y_d)^2}} \times \cdots \\
&\quad\quad e^{\frac{j2\pi}{\lambda} f \left(1 + \frac{(x - x_d)^2 + (y - y_d)^2}{2f^2}\right)} e^{\frac{-j2\pi(x^2 + y^2)}{2\lambda f}} dxdy \\
&= \quad \frac{e^{\frac{j2\pi}{\lambda} f}}{j\lambda} \int_{x_l - \frac{\Delta}{2}}^{x_l + \frac{\Delta}{2}} \int_{y_l - \frac{\Delta}{2}}^{y_l + \frac{\Delta}{2}} \frac{U(x_l, y_l) f}{f^2 + (x - x_d)^2 + (y - y_d)^2} e^{\frac{j2\pi}{\lambda} f \left(\frac{(x - x_d)^2 + (y - y_d)^2}{2f^2}\right)} e^{\frac{-j2\pi(x^2 + y^2)}{2\lambda f}} dxdy
\end{aligned} \tag{B.8}$$

Using the assumption that the propagation distance is much larger than the extent of the difference between the lens and detector coordinates allows the following approximation

$$\frac{f}{f^2+(x-x_d)^2+(y-y_d)^2} \approx \frac{f}{f^2+(x_l-x_d)^2+(y_l-y_d)^2} \equiv \Psi \tag{B.9}$$

which provides a discrete ratio that is independent of continuous (x, y). This ratio, denoted Ψ, is substituted into equation B.8 resulting in

$$\begin{aligned}
U_{LS}(x_d, y_d) \quad &\approx \quad \frac{e^{\frac{j2\pi}{\lambda}f}}{j\lambda} \int_{x_l-\frac{\Delta}{2}}^{x_l+\frac{\Delta}{2}} \int_{y_l-\frac{\Delta}{2}}^{y_l+\frac{\Delta}{2}} \Psi \; U(x_l, y_l) \; e^{j2\pi\left(\frac{(x-x_d)^2+(y-y_d)^2}{2\lambda f}\right)} e^{\frac{-j2\pi(x^2+y^2)}{2\lambda f}} dxdy \\
&\approx \quad \frac{\Psi \; e^{\frac{j2\pi}{\lambda}f}}{j\lambda} \int_{x_l-\frac{\Delta}{2}}^{x_l+\frac{\Delta}{2}} \int_{y_l-\frac{\Delta}{2}}^{y_l+\frac{\Delta}{2}} U(x_l, y_l) \; e^{\frac{j2\pi}{2\lambda f}\left((x-x_d)^2+(y-y_d)^2-(x^2+y^2)\right)} dxdy \\
&\approx \quad \frac{\Psi \; e^{\frac{j2\pi}{\lambda}f}}{j\lambda} \int_{x_l-\frac{\Delta}{2}}^{x_l+\frac{\Delta}{2}} \int_{y_l-\frac{\Delta}{2}}^{y_l+\frac{\Delta}{2}} U(x_l, y_l) \; e^{\frac{j2\pi}{2\lambda f}\left(x^2-2xx_d+x_d^2+y^2-2yy_d+y_d^2-x^2-y^2)\right)} dxdy \\
&\approx \quad \frac{\Psi}{j\lambda} e^{\frac{j2\pi}{\lambda}f} e^{\frac{j\pi}{\lambda f}(x_d^2+y_d^2)} \int_{x_l-\frac{\Delta}{2}}^{x_l+\frac{\Delta}{2}} \int_{y_l-\frac{\Delta}{2}}^{y_l+\frac{\Delta}{2}} U(x_l, y_l) \; e^{\frac{-j2\pi}{\lambda f}(xx_d+yy_d)} dxdy \\
&\approx \quad \frac{\Psi}{j\lambda} e^{\frac{j2\pi}{\lambda}f} e^{\frac{j\pi}{\lambda f}(x_d^2+y_d^2)} U(x_l, y_l) \int_{x_l-\frac{\Delta}{2}}^{x_l+\frac{\Delta}{2}} e^{\frac{-j2\pi}{\lambda f}(xx_d)} dx \int_{y_l-\frac{\Delta}{2}}^{y_l+\frac{\Delta}{2}} e^{\frac{-j2\pi}{\lambda f}(yy_d)} dy \\
&\approx \quad \frac{\Psi}{j\lambda} e^{\frac{j2\pi}{\lambda}f} e^{\frac{j\pi}{\lambda f}(x_d^2+y_d^2)} U(x_l, y_l) \left[\frac{-\lambda f}{j2\pi x_d} \; e^{\frac{-j2\pi}{\lambda f}(xx_d)} \Bigg|_{x \; = \; x_l-\frac{\Delta}{2}}^{x \; = \; x_l+\frac{\Delta}{2}} \right] \times \cdots \\
&\qquad \left[\frac{-\lambda f}{j2\pi y_d} \; e^{\frac{-j2\pi}{\lambda f}(yy_d)} \Bigg|_{y \; = \; y_l-\frac{\Delta}{2}}^{y \; = \; y_l+\frac{\Delta}{2}} \right] \\
&\approx \quad \frac{\Psi}{j\lambda} e^{\frac{j2\pi}{\lambda}f} e^{\frac{j\pi}{\lambda f}(x_d^2+y_d^2)} U(x_l, y_l) \left[\frac{-\lambda f}{j2\pi x_d} \; e^{\frac{-j2\pi x_l x_d}{\lambda f}} \left(e^{\frac{j\pi\Delta x_d}{\lambda f}} - e^{\frac{-j\pi\Delta x_d}{\lambda f}} \right) \right] \times \cdots \\
&\qquad \left[\frac{-\lambda f}{j2\pi y_d} \; e^{\frac{-j2\pi y_l y_d}{\lambda f}} \left(e^{\frac{j\pi\Delta y_d}{\lambda f}} - e^{\frac{-j\pi\Delta y_d}{\lambda f}} \right) \right] \\
&\approx \quad \frac{\Psi}{j\lambda} e^{\frac{j2\pi}{\lambda}f} e^{\frac{j\pi}{\lambda f}(x_d^2+y_d^2)} U(x_l, y_l) \left[\frac{-\lambda f}{j2\pi x_d} \; e^{\frac{-j2\pi x_l x_d}{\lambda f}} \left\{ -2j \sin\left(\frac{\pi\Delta x_d}{\lambda f}\right) \right\} \right] \times \cdots \\
&\qquad \left[\frac{-\lambda f}{j2\pi y_d} \; e^{\frac{-j2\pi y_l y_d}{\lambda f}} \left\{ -2j \sin\left(\frac{\pi\Delta y_d}{\lambda f}\right) \right\} \right] \\
&\approx \quad \frac{\Psi}{j\lambda} e^{\frac{j2\pi}{\lambda}f} e^{\frac{j\pi}{\lambda f}(x_d^2+y_d^2)} U(x_l, y_l) \left[e^{\frac{-j2\pi x_l x_d}{\lambda f}} \frac{\sin\frac{\pi\Delta x_d}{\lambda f}}{\frac{\pi x_d}{\lambda f}} \right] \left[e^{\frac{-j2\pi y_l y_d}{\lambda f}} \frac{\sin\frac{\pi\Delta y_d}{\lambda f}}{\frac{\pi y_d}{\lambda f}} \right]
\end{aligned}$$

(*continued overpage*)

$$\begin{aligned} U_{LS}(x_d, y_d) &\approx \frac{\Psi}{j\lambda} e^{\frac{j2\pi}{\lambda} f} e^{\frac{j\pi}{\lambda f}(x_d^2+y_d^2)} U(x_l, y_l) \left[e^{\frac{-j2\pi x_l x_d}{\lambda f}} \frac{\Delta \sin \frac{\pi \Delta x_d}{\lambda f}}{\frac{\Delta \pi x_d}{\lambda f}} \right] \left[e^{\frac{-j2\pi y_l y_d}{\lambda f}} \frac{\Delta \sin \frac{\pi \Delta y_d}{\lambda f}}{\frac{\Delta \pi y_d}{\lambda f}} \right] \\ &\approx \frac{\Psi}{j\lambda} e^{\frac{j2\pi}{\lambda} f} e^{\frac{j\pi}{\lambda f}(x_d^2+y_d^2)} U(x_l, y_l) \Delta^2 \left[e^{\frac{-j2\pi x_l x_d}{\lambda f}} sinc\left(\frac{\pi \Delta x_d}{\lambda f}\right) \right] \left[e^{\frac{-j2\pi y_l y_d}{\lambda f}} sinc\left(\frac{\pi \Delta y_d}{\lambda f}\right) \right] \\ &\approx \frac{\Psi}{j\lambda} e^{\frac{j2\pi}{\lambda} f} e^{\frac{j\pi}{\lambda f}(x_d^2+y_d^2)} \Delta^2 sinc\left(\frac{\pi \Delta x_d}{\lambda f}\right) sinc\left(\frac{\pi \Delta y_d}{\lambda f}\right) U(x_l, y_l)\ e^{\frac{-j2\pi x_l x_d}{\lambda f}} e^{\frac{-j2\pi y_l y_d}{\lambda f}} \end{aligned} \tag{B.10}$$

The field resulting from all lens samples is then the field sum of the contribution from each sample. This is represented as the sum of equation B.10 over the index to each lens sample in accordance with

$$\begin{aligned} U(x_d, y_d) &\approx \sum_{n=1}^{N} \sum_{m=1}^{M} U_{LS}(x_d, y_d) \\ &\approx \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{\Psi}{j\lambda} e^{\frac{j2\pi f}{\lambda}} e^{\frac{j\pi}{\lambda f}(x_d^2+y_d^2)} \Delta^2 sinc\left(\frac{\pi \Delta x_d}{\lambda f}\right) sinc\left(\frac{\pi \Delta y_d}{\lambda f}\right) \times \cdots \\ &\quad U(x_n, y_m)\ e^{\frac{-j2\pi x_n x_d}{\lambda f}} e^{\frac{-j2\pi y_m y_d}{\lambda f}} \end{aligned} \tag{B.11}$$

where the subscripts on $U(x_n, y_m)$ combine to index over all lens samples.

In order to simplify equation B.11 consider the following grouping of the exponential terms

$$\begin{aligned} e^{\frac{j2\pi f}{\lambda}} e^{\frac{j\pi(x_d^2+y_d^2)}{\lambda f}} e^{\frac{-j2\pi x_n x_d}{\lambda f}} e^{\frac{-j2\pi y_m y_d}{\lambda f}} &= e^{\frac{j2\pi f}{\lambda}} e^{\frac{j\pi(x_d^2+y_d^2)}{\lambda f}} e^{\frac{-j2\pi x_n x_d}{\lambda f}} e^{\frac{-j2\pi y_m y_d}{\lambda f}} e^{\frac{j2\pi(x_n^2+y_m^2)}{2\lambda f}} e^{\frac{-j2\pi(x_n^2+y_m^2)}{2\lambda f}} \\ &= e^{\frac{-j2\pi(x_n^2+y_m^2)}{2\lambda f}} e^{\frac{j2\pi f}{\lambda}} e^{\frac{j2\pi((x_n-x_d)^2+(y_m-y_d)^2)}{2\lambda f}} \\ &= e^{\frac{-j2\pi(x_n^2+y_m^2)}{2\lambda f}} e^{\frac{j2\pi}{\lambda}\left(f+\frac{(x_n-x_d)^2+(y_m-y_d)^2}{2f}\right)} \\ &= e^{\frac{-j2\pi(x_n^2+y_m^2)}{2\lambda f}} e^{\frac{j2\pi f}{\lambda}\left(1+\frac{(x_n-x_d)^2+(y_m-y_d)^2}{2f^2}\right)} \\ &= e^{\frac{-j2\pi(x_n^2+y_m^2)}{2\lambda f}} e^{\frac{j2\pi f}{\lambda}\sqrt{1+\frac{(x_n-x_d)^2+(y_m-y_d)^2}{f^2}}} \\ &= e^{\frac{-j2\pi(x_n^2+y_m^2)}{2\lambda f}} e^{\frac{j2\pi\sqrt{f^2+(x_n-x_d)^2+(y_m-y_d)^2}}{\lambda}} \\ &= t_l(x_n, y_m) e^{\frac{j2\pi\sqrt{f^2+(x_n-x_d)^2+(y_m-y_d)^2}}{\lambda}} \end{aligned} \tag{B.12}$$

The simplification in equation B.12 is substituted into equation B.11 as

$$\begin{aligned} U(x_d, y_d) &\approx \sum_{n=1}^{N}\sum_{m=1}^{M} \frac{\Psi}{j\lambda} e^{\frac{j2\pi f}{\lambda}} e^{\frac{j\pi}{\lambda f}(x_d^2+y_d^2)} \Delta^2 sinc\left(\frac{\pi \Delta x_d}{\lambda f}\right) sinc\left(\frac{\pi \Delta y_d}{\lambda f}\right) \times \cdots \\ & \quad U(x_n, y_m)\; e^{\frac{-j2\pi x_n x_d}{\lambda f}} e^{\frac{-j2\pi y_m y_d}{\lambda f}} \\ &\approx \frac{\Delta^2}{j\lambda} \sum_{n=1}^{N}\sum_{m=1}^{M} \Psi\; sinc\left(\frac{\pi \Delta x_d}{\lambda f}\right) sinc\left(\frac{\pi \Delta y_d}{\lambda f}\right) U(x_n, y_m)\; t_l(x_n, y_m) e^{\frac{j2\pi \sqrt{f^2+(x_n-x_d)^2+(y_m-y_d)^2}}{\lambda}} \end{aligned} \tag{B.13}$$

The discrete propagation distance Ψ at equation B.9 between lens samples and detector pixels is then substituted into equation B.13 to determine the final form of the discrete Rayleigh-Sommerfeld propagation expression as

$$U(x_d, y_d) \approx \frac{f\Delta^2}{j\lambda} \sum_{n=1}^{N}\sum_{m=1}^{M} \frac{U(x_n, y_m) t_l(x_n, y_m)\; e^{\frac{j2\pi}{\lambda}\sqrt{f^2+(x_n-x_d)^2+(y_m-y_d)^2}}}{f^2+(x_n-x_d)^2+(y_m-y_d)^2} sinc\left(\frac{\pi \Delta x_d}{f\lambda}\right) sinc\left(\frac{\pi \Delta y_d}{f\lambda}\right) \tag{B.14}$$

Also note that the intensity at the detector, $I(x_d, y_d)$, is the complex square of the electromagnetic field given by

$$I(x_d, y_d) = |U(x_d, y_d)|^2 = U(x_d, y_d) \times U^*(x_d, y_d) \tag{B.15}$$

when propagation calculations result in flux units of $\frac{photons}{second}$ as used in this thesis (as opposed to using flux units of $Watts$).

Appendix C. Matlab® Code

C.1 Summary of Matlab® files

This section provides a summary the Matlab® files developed to conduct the test cases in chapter IV.

Table C.1: Matlab® Files For Generating Hyperspectral Data Cubes

Matlab® File	**Test Case Scenario**	**Test Case Number**
make binary atm	Binary Star Pair	4.1.1, 4.1.2, 4.1.3, 4.1.4
make slabs 1b	Spatially Separate Monochromatic Source	4.2.1, 4.2.2, 4.2.3 4.2.4, 4.2.5, 4.2.6
make slabs 2	Spatially Overlapping Monochromatic Source	4.3.1, 4.3.2, 4.3.3 4.3.4, 4.3.5, 4.3.6
make number cube	Monochromatic Numbers Source	4.4.1, 4.4.2, 4.4.3
make fireball atm 2	Uniform Fireball	4.5.1.1, 4.5.1.2, 4.5.1.3 4.5.1.4, 4.5.1.5, 4.5.1.6
make fireball rings atm 2	Concentric Temperature Rings Fireball	4.5.2.1, 4.5.2.2 4.5.2.3, 4.5.2.4
make fireball spots atm 2	Hot-Spot Fireball	4.5.3.1, 4.5.3.2 4.5.3.3, 4.5.3.4
make fireball atm series	Evolving Fireballs	4.6.1, 4.6.2, 4.6.3, 4.6.4

Table C.2: Matlab® Files For Generating CTIS Detector Images

Matlab® File	**Test Case Scenario**	**Test Case Number**
At dectr 15 bins	All static test cases	4.1.1 to 4.5.3.4
At dectr evolve 15 bins	Evolving Fireballs	4.6.1, 4.6.2, 4.6.3, 4.6.4

Table C.3: Matlab® Files For 2D And Vector Reconstruction Algorithms

Matlab® File	**Test Case Scenario**	**Test Case Number**
Recon 2D no atm stars Recon 2D atm stars Recon vector no atm 15 stars Recon vector atm 15 stars	Binary Star Pair	4.1.1 4.1.2 4.1.3 4.1.4
Recon 2D no atm big bars Recon vector big scene	Both Spatially Separate and Spatially Overlapping Monochromatic Sources	4.2.1, 4.2.2 4.3.1, 4.3.2 4.2.3, 4.2.4, 4.2.5, 4.2.6 4.3.3, 4.3.4, 4.3.5, 4.3.6
Recon 2D no atm numbers Recon 2D atm numbers	Monochromatic Numbers Source	4.4.1, 4.4.3 4.4.2
Recon 2D no atm fball rings1 Recon 2D atm fball rings1 Recon vector 15 uni fball Recon vector 15 atm combo	Static Fireballs	4.5.2.1, 4.5.3.1 4.5.2.2, 4.5.3.2 4.5.1.1, 4.5.1.3, 4.5.1.5 4.5.2.3, 4.5.3.3 4.5.1.2, 4.5.1.4, 4.5.1.6 4.5.2.4, 4.5.3.4
Recon vector 15 evolve fball	Evolving Fireballs	4.6.1, 4.6.2, 4.6.3, 4.6.4

Table C.4: Miscellaneous CTIS Matlab® Files

Matlab® File	**Purpose**
Atmo calculator	Calculates the atmospheric transmission coefficient for each wavelength bin
Index inter 3	Calculates the spectral refractive indices for the DVP optical materials
Make OTF dectr no fftshift	Calculates the 2D OTF of the modelled CTIS
Make OTF dectr vector	Calculates the 1D OTF of the modelled CTIS
Prism rays 3	Calculates the radial spectral shift produced by the DVP
PSFs RS fn	Calculates the unshifted spectral PSF for light at the center wavelength at each bin
Results *(various)*	Presents the results for each reconstruction data run
Atmo data 15 bins	Data file of atmospheric transmission coefficients
at dectr*(various)*	Data file of the CTIS detector images produced by “At dectr 15 bins” and “At dectr evolve 15 bins”
bin data 15	Data file of the parameters for each wavelength bin
data run*(various)*	Data file of the output of the 2D and vector reconstruction algorithms in Table C.3
DVP angles 15	Data file of the DVP rotation angles used in the simulations
otf dectr*(various)*	Data file of the 2D OTF produced by “Make OTF dectr no fftshift”
OTF15 vectors *(various)*	Data file of OTF vectors produced by “Make OTF dectr vector”
photons*(various)*	Data file of the output from hyperspectral data cube generating files in Table C.1
PSF unshifted 15	Data file of the unshifted spectral PSFs produced by “PSFs RS fn”
Shifts*(various)*	Data file of the cartesian spectral shifts
Pixel temperature fit*(various)*	Temperature fitting to estimated hyperspectral data cubes produced by 2D and vector reconstruction algorithms

C.2 Matlab® Files For Generating Hyperspectral Data Cubes

C.2.0.1 make_binary_atm.m. This file generates the hyperspectral data cubes for the binary star pair.

```
clear all
% Fundamental constants.
c = 299792458;
light_year = c * 60 * 60 * 24 * 365; % units of meters
h = 6.6260755e-34;
kB = 1.380658e-23;

% Atmospheric coefficient data
load Atmo_data_15
%load Atmo_data_50

% Number of bins being used and bin parameters for
% selected wavelength range of 2 to 5 microns
bins = length(atmo_av_trans);
int = (5-2)/bins;
lamda = 2:int:5-int;
bin_low_cent_high = [lamda' lamda'+int/2 lamda'+int];

% Number of pixels along a side of the true image
% This limits the spatial extent of the object image cube
N = 20; %100;
x = -N/2:N/2-1;
[X,Y] = meshgrid(x,x);

% True temperature of object cube in units of degrees kelvin
truth = zeros(N) + 10;
% Background temperature of space is listed at 3 degrees Kelvin
% Create random background temperature from a Poisson RV  with mean 10
%truth = zeros(N) + poissrnd(10,N,N);
truth(11,7) = 10000;   % Star 1 has a temperature of 5000 K
truth(11,14) = 5000;   % Star 2 has a temperature of 10000 K
figure(1), imagesc(truth), colorbar('vert'), axis square xy

% Gives the spectral photon radiance [photons/sec - cm^2 - sr - um] as a
% sum of "sub_lamda" sub-bin areas in each wavelength bin.
sub_lamda = 100;
for index = 1:bins
    lamda_base = lamda(index):int/sub_lamda:lamda(index) + int - int/sub_lamda;
    temporary = zeros(N);
    for count = 1:sub_lamda
        temporary = temporary + (2 * c * 1e14) ./ (lamda_base(count)^4 .* ...
            (exp(h*c./(lamda_base(count)*1e-6*kB.*truth)) - 1)) * int/sub_lamda;
    end
    rad_p(index,:,:) = temporary;
end
%figure(1), for k = 1:15, temp = squeeze(rad_p(k,:,:));
%imagesc(temp), colorbar('vert'), axis square xy,  pause(.2), end

sol_rad =  6.96e10; % Solar radii in units of centimetres % 1 solar radii = 6.96e8 meters
R1 = 2.5 ; %.25; % Radius of star 1 in units of Solar Radii;
R2 = 1.1 ; %5;    % Radius of star 2 in units of Solar Radii;
A1 = pi*(R1*sol_rad)^2;  % Projected area of star1 in centimetres
```

```
A2 = pi*(R2*sol_rad)^2;  % Projected area of star2 in centimetres
Dist = 50; % Light years
D = .1; % diameter of the telescope in meters;
SR1 = (D/2)^2*pi/(Dist * light_year)^2; % Solid angle subtended by  star 1 to sensor
SR2 = (D/2)^2*pi/(Dist * light_year)^2; % Solid angle subtended by star 2 to sensor

dt=.001; % Integration time in seconds of the CCD camera

% Calulates the photon flux at detector using Flux = Rad_p * Area of source * Solid angle
% ie, (Lp*As*Ad/r) where Lp = Rad_p, As = Area of source, Ad = Area of detector
% and r = distance between source and detector
for index = 1 :bins
    temp = squeeze(rad_p(index,:,:));
    photons(index,:,:) = [temp(:,1:10)*dt*A1*SR1 temp(:,11:20)*dt*A2*SR2];
end

% This loop creates x & y vector sums of the source photons with no atmospheric attenuation
for k = 1:bins
    photons_x(k,:) = sum(squeeze(photons(k,:,:)));
    photons_y(k,:) = sum(squeeze(photons(k,:,:))');
end

% This loop attenuates photons by atmospheric transimission coefficients
%for index = 1:bins, photons(index,:,:) = photons(index,:,:) * atmo_av_trans(index); end

% Physical photons exists in integer quantities
photons = round(photons);  %Rounds #photons to an integer

%figure(1), for k = 1:15, temp = squeeze(photons(k,:,:));
%imagesc(temp), colorbar('vert'), axis square xy,  pause(.2), end
h = 1;
figure(1), for k = [1 5 10 15],
    temp = squeeze(photons(k,:,:));
    subplot(2,2,h), imagesc(temp), h = h+1;
    title(['Photons @ \lambda = ' num2str(bin_low_cent_high(k,2)) '\mum'])
    colorbar('vert'), axis square xy,
end
figure(2)
for k = 1:bins, temp = squeeze(photons(k,:,:)); star1(k) = sum(sum(temp(:,1:10)));
    star2(k) = sum(sum(temp(:,11:20))); end
subplot(211), plot(bin_low_cent_high(:,2), star1,'r+'), legend('Original Star 1 @ T = 10000K')
xlabel('Wavelength - \mum'), ylabel('Number of Photons')
subplot(212), plot(bin_low_cent_high(:,2), star2,'rx'), legend('Original Star 2 @ T = 5000K')
xlabel('Wavelength - \mum'), ylabel('Number of Photons')

whos
```

C.2.0.2 make_slabs_1b. This file generates the hyperspectral data cubes for the spatially separate monochromatic source.

```
clear all
N = 20;
bins = 15;
back = zeros(N);
photons = zeros(bins,N,N);

temp = back;
temp(3:18,2:4) = 5;
photons(1,:,:) = temp;

temp = back;
temp(3:18,17:19) = 5;
photons(4,:,:) = temp;

temp = back;
temp(3:5,7:14) = 5;
photons(8,:,:) = temp;

temp = back;
temp(9:12,7:14) = 5;
photons(12,:,:) = temp;

temp = back;
temp(16:18,7:14) = 5;
photons(15,:,:) = temp;

for k = 1:bins
    temp = squeeze(photons(k,:,:));
    photonb(k,:,:)  = imresize(temp,5);
end
photons = photonb;

for k = 1:bins
    photons_x(k,:) = sum(squeeze(photons(k,:,:)));
    photons_y(k,:) = sum(squeeze(photons(k,:,:))');
end
```

C.2.0.3 make_slabs_2. This file generates the hyperspectral data cubes for the spatially overlapping monochromatic source.

```
clear all
N = 20;
bins = 15;
back = zeros(N);
photons = zeros(bins,N,N);

temp = back;
temp(3:18,2:6) = 5;
photons(1,:,:) = temp;

temp = back;
temp(3:18,15:19) = 5;
photons(5,:,:) = temp;

temp = back;
temp(3:7,3:18) = 5;
photons(10,:,:) = temp;

temp = back;
temp(14:18,3:18) = 5;
photons(15,:,:) = temp;

for k = 1:bins
    temp = squeeze(photons(k,:,:));
    photonb(k,:,:)  = imresize(temp,5);
end
photons = photonb;

for k = 1:bins
    photons_x(k,:) = sum(squeeze(photons(k,:,:)));
    photons_y(k,:) = sum(squeeze(photons(k,:,:))’);
end
```

C.2.0.4 make_number_cube. This file generates the hyperspectral data cubes for the spatially separate monochromatic source.

```
clear all
level = 100;
open one.jpg, temp = one;
open two.jpg, temp = two;
open three.jpg, temp = three;
open four.jpg, temp = four;
open five.jpg, temp = five;
open six.jpg, temp = six;
open seven.jpg, temp = seven;
open eight.jpg, temp = eight;
open nine.jpg, temp = nine;
open ten.jpg, temp = ten;
open eleven.jpg, temp = eleven;
open twelve.jpg, temp = twelve;
open thirteen.jpg, temp = thirteen;
open fourteen.jpg, temp = fourteen;
open fifteen.jpg, temp = fifteen;

temp=sum(temp,3);
temp = level * (temp<=500);
figure(1), imagesc(temp), colorbar('vert')

NC1 = temp;
NC2 = temp;
NC3 = temp;
NC4 = temp;
NC5 = temp;
NC6 = temp;
NC7 = temp;
NC8 = temp;
NC9 = temp;
NC10 = temp;
NC11 = temp;
NC12 = temp;
NC13 = temp;
NC14 = temp;
NC15 = temp;

photons(1,:,:) = flipud(NC1);
photons(2,:,:) = flipud(NC2);
photons(3,:,:) = flipud(NC3);
photons(4,:,:) = flipud(NC4);
photons(5,:,:) = flipud(NC5);
photons(6,:,:) = flipud(NC6);
photons(7,:,:) = flipud(NC7);
photons(8,:,:) = flipud(NC8);
photons(9,:,:) = flipud(NC9);
photons(10,:,:) = flipud(NC10);
photons(11,:,:) = flipud(NC11);
photons(12,:,:) = flipud(NC12);
photons(13,:,:) = flipud(NC13);
photons(14,:,:) = flipud(NC14);
photons(15,:,:) = flipud(NC15);

for k = 1:15
    temp1 = squeeze(photons(k,:,:));
    figure(1), subplot(111)
    imagesc(temp1), axis square xy
    pause(0.5)
end

for k = 1:15
    photons_x(k,:) = sum(squeeze(photons(k,:,:)));
    photons_y(k,:) = sum(squeeze(photons(k,:,:))');
end

whos
```

C.2.0.5 make_fireball_atm_2. This file generates the hyperspectral data cubes for the uniform fireballs.

```
clear all
tic

% Fundamental constants.
c = 299792458;
h = 6.6260755e-34;
kB = 1.380658e-23;

load Atmo_data_15

N = 100;  % number of pixels along a side of the true image
x = -N/2:N/2-1;
[X,Y] = meshgrid(x,x);
R = sqrt(X.^2 +Y.^2);
ball = (R<=5); figure(1), subplot(111), imagesc(ball)
back = (ball~=1); figure(1), imagesc(back)
%truth = zeros(N,N)+10;  %units of degrees kelvin
%truth = ball * 1600 + back .* normrnd(300,10,N,N); %units of degrees kelvin
%truth = ball * 1000 + back .* normrnd(300,10,N,N); %units of degrees kelvin
truth = ball * 400 + back .* normrnd(300,10,N,N); %units of degrees kelvin
figure(1), imagesc(truth), colorbar('vert'), axis square xy
bins = 15;
int = (5-2)/bins;
lamda = 2:int:5-int;  % My selected wavelength range
bin_low_cent_high = [lamda' lamda'+int/2 lamda'+int];

% Gives the spectral photon radiance [photons/sec - cm^2 - sr - um] as a
% sum of "sub_lamda" sub-bin areas in each wavelength bin.
sub_lamda = 100;
for index = 1:bins
    lamda_base = lamda(index):int/sub_lamda:lamda(index) + int - int/sub_lamda;
    temporary = zeros(N,N);
    for count = 1:sub_lamda
        temporary = temporary + (2 * c * 1e14) ./ (lamda_base(count)^4 .* ...
            (exp(h*c./(lamda_base(count)*1e-6*kB.*truth)) - 1)) * int/sub_lamda;
    end
    rad_p(index,:,:) = temporary;
end

%Projected area of a square pixel at 3258m entered in metres^2 converted to centimetres^2
A1 = 3.258^2 * 1000;

D = .1; % diameter of the telescope in metres;
Dist = 3258; % Distance for fireball to sensor in metres
SR = D*D*pi/(4*Dist); % Solid angle subtended by sensor to fireball range

dt=.001; % Integration time in seconds of the CCD camera
%dt = .01;

for index = 1:bins
    photons(index,:,:) = squeeze(rad_p(index,:,:)) * dt * A1 * SR;
end
```

```
% This loop creates x & y vector sums of the source photons with no atmospheric attenuation
for k = 1:bins
    photons_x(k,:) = sum(squeeze(photons(k,:,:)));
    photons_y(k,:) = sum(squeeze(photons(k,:,:))');
end

% This loop adds in atmospheric attenuation
for index = 1:bins,  photons(index,:,:) = photons(index,:,:) * atmo_av_trans(index); end

photons = round(photons);  %Rounds #photons to an integer

%whos
toc
```

C.2.0.6 make_fireball_rings_atm_2. This file generates the hyperspectral data cubes for the concentric temperature rings fireball.

```
clear all
tic

% Fundamental constants.
c = 299792458;
h = 6.6260755e-34;
kB = 1.380658e-23;

load Atmo_data_15
%load Atmo_data_30

N = 100;  % number of pixels along a side of the true image
x = -N/2:N/2-1;
[X,Y] = meshgrid(x,x);
R = sqrt(X.^2 +Y.^2);
ball1 = (R<=5);
ball2 = (R>5 & R<=10);
ball3 = (R>10 & R<=15);
ball4 = (R>15 & R<=20);
ball5 = (R>20 & R<=25);
back = (ball1+ball2+ball3+ball4+ball5 ~=1); figure(1), imagesc(back)
%truth = zeros(N,N)+10;  %units of degrees kelvin
truth = ball1*1600+ball2*1500+ball3*1200+ball4*1000+ball5*800+ back ...
    .* normrnd(300,10,N,N); %units of degrees kelvin
figure(1), imagesc(truth), colorbar('vert'), axis square xy
bins = 15;
int = (5-2)/bins;
lamda = 2:int:5-int;  % My selected wavelength range
bin_low_cent_high = [lamda' lamda'+int/2 lamda'+int];

% Gives the spectral photon radiance [photons/sec - cm^2 - sr - um] as a
% sum of "sub_lamda" sub-bin areas in each wavelength bin.
sub_lamda = 100;
for index = 1:bins
    lamda_base = lamda(index):int/sub_lamda:lamda(index) + int - int/sub_lamda;
    temporary = zeros(N,N);
    for count = 1:sub_lamda
        temporary = temporary + (2 * c * 1e14) ./ (lamda_base(count)^4 .* ..
        (exp(h*c./(lamda_base(count)*1e-6*kB.*truth)) - 1)) * int/sub_lamda;
    end
    rad_p(index,:,:) = temporary;
end

%Projected area of a square pixel at 3258m entered in metres^2 converted to centimetres^2
A1 = 3.258^2 * 1000;
D = .1; % diameter of the telescope in metres;
Dist = 3258; % Distance for fireball to sensor in metres
SR = D*D*pi/(4*Dist); % Solid angle subtended by sensor to fireball range

dt=.001; % Integration time in seconds of the CCD camera
%dt = .01;

for index = 1:bins
    photons(index,:,:) = squeeze(rad_p(index,:,:)) * dt * A1 * SR;
end

% This loop creates x & y vector sums of the source photons with no atmospheric attenuation
for k = 1:bins
    photons_x(k,:) = sum(squeeze(photons(k,:,:)));
    photons_y(k,:) = sum(squeeze(photons(k,:,:))');
end

% This loop adds in atmospheric attenuation
for index = 1:bins,  photons(index,:,:) = photons(index,:,:) * atmo_av_trans(index); end

photons = round(photons);  %Rounds #photons to an integer

%whos
toc
```

C.2.0.7 make_fireball_spots_atm_2. This file generates the hyperspectral data cubes for the hot-spots fireball.

```
clear all
tic

% Fundamental constants.
c = 299792458;
h = 6.6260755e-34;
kB = 1.380658e-23;

load Atmo_data_15
%load Atmo_data_30

N = 100; % number of pixels along a side of the true image
x = -N/2:N/2-1;
[X,Y] = meshgrid(x,x);
R = sqrt(X.^2 +Y.^2);
ball1 = sqrt((X-10).^2 +(Y-10).^2); ball1 = (ball1<=4);
ball2 = sqrt((X-10).^2 +(Y+10).^2); ball2 = (ball2<=6);
ball3 = sqrt((X+10).^2 +(Y-10).^2); ball3 = (ball3<=8);
ball4 = sqrt((X+10).^2 +(Y+10).^2); ball4 = (ball4<=10);
ball5 = (R<=25) - ball1 - ball2 - ball3 - ball4;
back = (R >25); figure(1), imagesc(back)
%truth = zeros(N,N)+10;  %units of degrees kelvin
```

```
truth = ball1*1600+ball2*1500+ball3*1200+ball4*1000+ball5*600+back ...
    .* normrnd(300,10,N,N); %units of degrees kelvin
figure(1), imagesc(truth), colorbar('vert'), axis square xy
bins = 15;
int = (5-2)/bins;
lamda = 2:int:5-int;  % My selected wavelength range
bin_low_cent_high = [lamda' lamda'+int/2 lamda'+int];

% Gives the spectral photon radiance [photons/sec - cm^2 - sr - um] as a
% sum of "sub_lamda" sub-bin areas in each wavelength bin.
sub_lamda = 100;
for index = 1:bins
    lamda_base = lamda(index):int/sub_lamda:lamda(index) + int - int/sub_lamda;
    temporary = zeros(N,N);
    for count = 1:sub_lamda
        temporary = temporary + (2 * c * 1e14) ./ (lamda_base(count)^4 .* ...
            (exp(h*c./(lamda_base(count)*1e-6*kB.*truth)) - 1)) * int/sub_lamda;
    end
    rad_p(index,:,:) = temporary;
end

%Projected area of a square pixel at 3258m entered in metres^2 converted to centimetres^2
A1 = 3.258^2 * 1000;
D = .1; % diameter of the telescope in metres;
Dist = 3258; % Distance for fireball to sensor in metres
SR = D*D*pi/(4*Dist); % Solid angle subtended by sensor to fireball range

dt=.001; % Integration time in seconds of the CCD camera
%dt = .01;

for index = 1:bins
    photons(index,:,:) = squeeze(rad_p(index,:,:)) * dt * A1 * SR;
end

% This loop creates x & y vector sums of the source photons with no atmospheric attenuation
for k = 1:bins
    photons_x(k,:) = sum(squeeze(photons(k,:,:)));
    photons_y(k,:) = sum(squeeze(photons(k,:,:))');
end

% This loop adds in atmospheric attenuation
for index = 1:bins,  photons(index,:,:) = photons(index,:,:) * atmo_av_trans(index); end

photons = round(photons);  %Rounds #photons to an integer

%whos
toc
```

C.2.0.8 make_fireball_atm_series. This file generates the hyperspectral data cubes for the evolving fireballs.

```
clear all
tic

% Fundamental constants.
c = 299792458;
h = 6.6260755e-34;
kB = 1.380658e-23;

load Atmo_data_15

N = 100; % number of pixels along a side of the true image
x = -N/2:N/2-1;
[X,Y] = meshgrid(x,x);
R = sqrt(X.^2 +Y.^2);
ball = (R<=5); figure(1), subplot(111), imagesc(ball)
back = (ball~=1); figure(1), imagesc(back)
%truth = zeros(N,N)+10;  %units of degrees kelvin
%truth = ball * 1600 + back .* normrnd(300,10,N,N); %units of degrees kelvin
%truth = ball * 1500 + back .* normrnd(300,10,N,N); %units of degrees kelvin
%truth = ball * 1400 + back .* normrnd(300,10,N,N); %units of degrees kelvin
%truth = ball * 1300 + back .* normrnd(300,10,N,N); %units of degrees kelvin
%truth = ball * 1200 + back .* normrnd(300,10,N,N); %units of degrees kelvin
truth = ball * 1000 + back .* normrnd(300,10,N,N); %units of degrees kelvin
%truth = ball * 400 + back .* normrnd(300,10,N,N); %units of degrees kelvin
figure(1), imagesc(truth), colorbar(’vert’), axis square xy
bins = 15;
int = (5-2)/bins;
lamda = 2:int:5-int;  % My selected wavelength range
bin_low_cent_high = [lamda’ lamda’+int/2 lamda’+int];

% Gives the spectral photon radiance [photons/sec - cm^2 - sr - um] as a
% sum of "sub_lamda" sub-bin areas in each wavelength bin.
sub_lamda = 100;
for index = 1:bins
    lamda_base = lamda(index):int/sub_lamda:lamda(index) + int - int/sub_lamda;
    temporary = zeros(N,N);
    for count = 1:sub_lamda
        temporary = temporary + (2 * c * 1e14) ./ (lamda_base(count)^4 .*...
            (exp(h*c./(lamda_base(count)*1e-6*kB.*truth)) - 1)) * int/sub_lamda;
    end
    rad_p(index,:,:) = temporary;
end

%Projected area of a square pixel at 3258m entered in metres^2 converted to centimetres^2
A1 = 3.258^2 * 1000;
D = .1; % diameter of the telescope in metres;
Dist = 3258; % Distance for fireball to sensor in metres
SR = D*D*pi/(4*Dist); % Solid angle subtended by sensor to fireball range

dt=.001; % Integration time in seconds of the CCD camera
%dt = .01;
```

```
for index = 1:bins
    photons(index,:,:) = squeeze(rad_p(index,:,:)) * dt * A1 * SR;
end

% This loop creates x & y vector sums of the source photons with no atmospheric attenuation
for k = 1:bins
    photons_x(k,:) = sum(squeeze(photons(k,:,:)));
    photons_y(k,:) = sum(squeeze(photons(k,:,:))');
end

% This loop adds in atmospheric attenuation
%for index = 1:bins,  photons(index,:,:) = photons(index,:,:) * atmo_av_trans(index); end

photons = round(photons);  %Rounds #photons to an integer

%whos
toc
```

C.3 Matlab® Files For Generating CTIS Detector Images

C.3.0.1 At_Dectr_15_bins. This file generates the CTIS detector images for the static test cases.

```
clear all
tic

DS = 256;                %256 Pixels in detector side

load bin_data_15
%load bin_data_50
rot_angle = length(bin_low_cent_high); %15 %30; % Number of rotation angles
                                                % also equal to number of wavelength bins
Shifts = Prism_rays_3(bin_low_cent_high(:,2));

Prism_stops = (0: 2*pi / rot_angle : 2*pi - 2*pi / rot_angle);
r = Shifts(:,2) * 1e4;                % Scale factor on Zernike coefficient
                                      % for 100um pixel pitch use 1/100e-6 = 1e4
                                      % for 66.67um pixel pitch use 1/66.67e-6 = 1.5e4
for count = 1 : rot_angle
    m(:,count) = r * cos(Prism_stops(count));             % Zernike 2 coefficient
    n(:,count) = r * sin(Prism_stops(count));             % Zernike 3 coefficient
end
m = round(m)+DS/2+1;
n = round(n)+DS/2+1;

%PSF = PSFs_RS_fn(bin_low_cent_high(:,2));
load PSF_unshifted_15
%load PSF_unshifted_50

% Loads in spectral star data for 15 bins
%load('...Final Code\Binary Star\photons_bin_star_15')
%load('...Final Code\Binary Star\photons_bin_star_15_atm')
%load('...Final Code\Binary Star\photons_bin_star_50')
%load('...Final Code\Binary Star\photons_bin_star_50_atm')
%load('...Final Code\Monochromatic Bars\photons_slab1s')
```

```
%load('...Final Code\Monochromatic Bars\photons_slab2s')
%load('...Final Code\Monochromatic Bars\photons_slab1b')
load('...Final Code\Monochromatic Bars\photons_slab2b')
%load('...Final Code\Big Numbers\photons_numbers')
%load('...Final Code\Big Numbers\photons_numbers_atm')
%load('...Final Code\Big Numbers\photons_numbers_small')
%load('...Final Code\Fireball 1\photons_fball4K_15')
%load('...Final Code\Fireball 1\photons_fball4K_15_atm')
%load('...Final Code\Fireball 1\photons_fball10K_15')
%load('...Final Code\Fireball 1\photons_fball10K_15_atm')
%load('...Final Code\Fireball 1\photons_fball16K_15')
%load('...Final Code\Fireball 1\photons_fball16K_15_atm')
%load('...Final Code\Fireball Rings\photons_fball_rings1_15')
%load('...Final Code\Fireball Rings\photons_fball_rings1_15_atm')
%load('...Final Code\Fireball Spots\photons_fball_spots1_15')
%load('...Final Code\Fireball Spots\photons_fball_spots1_15_atm')

%%%% Caculates the image at detector from binary stars
for angle = 1 : rot_angle;
    dectr_back = zeros(DS);
    m1 = m(:,angle);
    n1 = n(:,angle);
    for shift =   1 : rot_angle;
        mm = m1(shift)-60:m1(shift)+59;  nn = n1(shift)-60:n1(shift)+59; % 100 X 100 image
%        mm = m1(shift)-20:m1(shift)+19;  nn = n1(shift)-20:n1(shift)+19; % 20 X 20 image
        dectr_back(mm,nn) = ...
         conv2(squeeze(PSF(shift,:,:)),squeeze(photons(shift,:,:))) + dectr_back(mm,nn);
    end
    at_dectr(angle,:,:) = dectr_back; % Image at the detector at each rotation angle
% Image at the detector at each rotation angle with Poisson noise
%    at_dectr(angle,:,:) = poissrnd(dectr_back);
end

% Calculates vector image at detector
for k = 1:rot_angle
    at_dectr_x(k,:) = sum(squeeze(at_dectr(k,:,:)));
    at_dectr_y(k,:) = sum(squeeze(at_dectr(k,:,:))');
%    photons_x(k,:) = sum(squeeze(photons(k,:,:)));
%    photons_y(k,:) = sum(squeeze(photons(k,:,:))');
end

whos
toc

%figure(1), subplot(111), for k = 1:rot_angle, temp = squeeze(at_dectr(k,:,:));
%imagesc(temp), colorbar('vert'), axis square xy, pause(.2), end

%figure(2), subplot(111), for k = 1:rot_angle, temp = at_dectr_x(k,:);
%plot(temp),  axis([1 DS 0 max(temp)]), pause(.2), end
```

C.3.0.2 At_Dectr_evolve_15_bins.m. This file generates the CTIS detector images for the evolving fireball test cases.

```
clear all

tic

DS = 256;                %256 Pixels in detector side

load bin_data_15
%load bin_data_50
rot_angle = length(bin_low_cent_high); %15 %30; % Number of rotation angles
                                                % also equal to number of wavelength bins
Shifts = Prism_rays_3(bin_low_cent_high(:,2));

Prism_stops = (0: 2*pi / rot_angle : 2*pi - 2*pi / rot_angle);
r = Shifts(:,2) * 1e4;         % Scale factor on Zernike coefficient
                               % for 100um pixel pitch use 1/100e-6 = 1e4
                               % for 66.67um pixel pitch use 1/66.67e-6 = 1.5e4
for count = 1 : rot_angle
    m(:,count) = r * cos(Prism_stops(count));      % Zernike 2 coefficient
    n(:,count) = r * sin(Prism_stops(count));      % Zernike 3 coefficient
end
m = round(m)+DS/2+1;
n = round(n)+DS/2+1;

%PSF = PSFs_RS_fn(bin_low_cent_high(:,2));
load PSF_unshifted_15
%load PSF_unshifted_50

% Loads in spectral star data for 15 bins
load('...Final Code\Fireball Evolve\photons_fball16K_15'); photons16 = photons;
load('...Final Code\Fireball Evolve\photons_fball15K_15'); photons15 = photons;
load('...Final Code\Fireball Evolve\photons_fball14K_15'); photons14 = photons;
load('...Final Code\Fireball Evolve\photons_fball13K_15'); photons13 = photons;
load('...Final Code\Fireball Evolve\photons_fball12K_15'); photons12 = photons;
load('...Final Code\Fireball Evolve\photons_fball10K_15'); photons10 = photons;
load('...Final Code\Fireball Evolve\photons_fball4K_15'); photons4 = photons;
clear photons

%%%% Caculates the image at detector from binary stars
for angle = 1 : rot_angle;
 %   if (angle >=1 & angle <8)
%       photons = photons13;
        %   else                 % Code for two fireball temperature detector image
%       photons = photons16;
        %    end
    if angle >=1 & angle <=5
        photons = photons16;
    elseif angle >= 6 & angle <=10
        photons = photons10;     % Code for three fireball temperature detector image
    elseif angle >= 11
        photons = photons4;
    end
    dectr_back = zeros(DS);
    m1 = m(:,angle);
    n1 = n(:,angle);
    for shift =   1 : rot_angle;
        mm = m1(shift)-60:m1(shift)+59;  nn = n1(shift)-60:n1(shift)+59; % 100 X 100 image
%        mm = m1(shift)-20:m1(shift)+19;  nn = n1(shift)-20:n1(shift)+19; % 20 X 20 image
        dectr_back(mm,nn) = ...
           conv2(squeeze(PSF(shift,:,:)),squeeze(photons(shift,:,:))) + dectr_back(mm,nn);
    end
    at_dectr(angle,:,:) = dectr_back;      % Image at the detector at each rotation angle
% Image at the detector at each rotation angle with Poisson noise
%    at_dectr(angle,:,:) = poissrnd(dectr_back);
end

% Calculates vector image at detector
for k = 1:rot_angle
    at_dectr_x(k,:) = sum(squeeze(at_dectr(k,:,:)));
    at_dectr_y(k,:) = sum(squeeze(at_dectr(k,:,:))');
%    photons_x(k,:) = sum(squeeze(photons(k,:,:)));
%    photons_y(k,:) = sum(squeeze(photons(k,:,:))');
end

%whos
toc

%figure(1), subplot(111), for k = 1:rot_angle, temp = squeeze(at_dectr(k,:,:));
%imagesc(temp), colorbar('vert'), axis square xy, pause(.2), end

%figure(2), subplot(111), for k = 1:rot_angle, temp = at_dectr_x(k,:); plot(temp),
%axis([1 DS 0 max(temp)]), pause(.2), end
```

C.4 Matlab® Files For 2D And Vector Reconstruction Algorithms

C.4.0.1 Recon_2D_no_atm_stars. This file shows an example of the implementation of the 2D reconstruction algorithm.

```
clear all;
load('...Final Code\BinaryStar\at_dectr_bin_star_15')
load('...Final Code\Binary Star\photons_bin_star_15')

load Shifts_15_1o5e
load PSF_unshifted_15
load bin_data_15

rot_angle = 15;
DS = 256;

data_cube = at_dectr;

recon = ones(rot_angle,DS,DS);

for iters = 1:100
    tic
    est_data1 = zeros(rot_angle,DS,DS);
    for angle = 1:rot_angle;
        load(['...Final Code\OTF15 no fftshift\otf_dectrA_' num2str(angle)])

        temp6 = zeros(DS);
        for shift = 1:rot_angle;
            temp4 = (fft2(squeeze(recon(shift,:,:))));
            temp5 = temp4 .* squeeze(otf_dectr(shift,:,:));
            temp6 = real(ifft2((temp5))) + temp6;
        end
        est_data1(angle,:,:) = (temp6);
        clear otf_dectr
    end
    differ = sum(sum(sum(abs(est_data1-data_cube))))
    MSE(iters) = differ;
    map = (est_data1==0);
    est_data1 = (1-map) .* data_cube ./ (est_data1 + map) + map;

    for shift = 1:rot_angle;
        load(['...Final Code\OTF15 no fftshift\otf_dectrB_' num2str(shift)])

        temp1 = zeros(DS);
        for angle = 1:rot_angle;
            temp2 = (fft2(squeeze(est_data1(angle,:,:))));
            temp3 = temp2 .* conj(squeeze(otf_dectr(angle,:,:)));
            temp1 = real(ifft2((temp3))) + temp1;
        end
        est_data2(shift,:,:) = (temp1);
        clear otf_dectr
    end

    recon = recon .* est_data2 / rot_angle;
    count = iters
    it_time = toc
```

```
end

for shift = 1:15
    recon(shift,:,:) = fftshift(squeeze(recon(shift,:,:)));
end

for k = 1:15;
    temp =abs(squeeze(recon(k,:,:)));
    photon_temp = squeeze(photons(k,:,:));
    star1_r(k) = round(sum(sum(temp(:,1:128))));
    star2_r(k) = round(sum(sum(temp(:,129:256))));
    star1(k) = round(sum(sum(photon_temp(:,1:10))));
    star2(k) = round(sum(sum(photon_temp(:,11:20))));
    figure(2), subplot(111),
    imagesc(temp), axis([119 138 119 138]), axis square, axis xy
    colorbar('vert'), title(['bin = ' num2str(k)]), pause(.01)
end

figure(1)
subplot(211), plot(bin_low_cent_high(:,2), star1,'r+',bin_low_cent_high(:,2), star1_r,'bo'),
legend('Original Star 1 @ T = 10000K', 'Reconstructed Star 1')
xlabel('Wavelength - \mum'), ylabel('Number of Photons')
subplot(212), plot(bin_low_cent_high(:,2), star2,'rx',bin_low_cent_high(:,2), star2_r,'gd')
legend('Original Star 2 @ T = 5000K','Reconstructed Star 2')
xlabel('Wavelength - \mum'), ylabel('Number of Photons')

C = [star1_r' star1' star1_r'./star1'  star2_r' star2' star2_r'./star2']

whos
toc
```

C.4.0.2 Recon_2D_atm_stars. This file shows an example of the implementation of the 2D atmospheric reconstruction algorithm.

```
clear all;
load('...Final Code\Binary Star\at_dectr_bin_star_15_atm')
load('...Final Code\Binary Star\photons_bin_star_15')

load Atmo_data_15
atmo = atmo_av_trans;
T = sum(atmo);

load Shifts_15_1o5e
load PSF_unshifted_15
load bin_data_15

rot_angle = 15;
DS = 256;

data_cube = at_dectr;

recon = ones(rot_angle,DS,DS);

for iters = 1:100
    tic
    est_data1 = zeros(rot_angle,DS,DS);
    for angle = 1:rot_angle;
        load(['...Final Code\OTF15 no fftshift\otf_dectrA_' num2str(angle)])

        temp6 = zeros(DS);
        for shift = 1:rot_angle;
            temp4 = (fft2(squeeze(recon(shift,:,:))));
            temp5 = temp4 .* squeeze(otf_dectr(shift,:,:)) * atmo(shift);
            temp6 = real(ifft2((temp5))) + temp6;
        end
        est_data1(angle,:,:) = (temp6);
        clear otf_dectr
    end
    differ = sum(sum(sum(abs(est_data1-data_cube))))
    MSE(iters) = differ;
    map = (est_data1==0);
    est_data1 = (1-map) .* data_cube ./ (est_data1 + map) + map;

    for shift = 1:rot_angle;
        load(['...Final Code\OTF15 no fftshift\otf_dectrB_' num2str(shift)])

        temp1 = zeros(DS);
        for angle = 1:rot_angle;
            temp2 = (fft2(squeeze(est_data1(angle,:,:))));
            temp3 = temp2 .* conj(squeeze(otf_dectr(angle,:,:))) * atmo(angle);
            temp1 = real(ifft2((temp3))) + temp1;
        end
        est_data2(shift,:,:) = (temp1);
        clear otf_dectr
    end

    recon = recon .* est_data2 / T;
    count = iters
    it_time = toc

end

for shift = 1:15
    recon(shift,:,:) = fftshift(squeeze(recon(shift,:,:)));
end

for k = 1:15;
    temp =abs(squeeze(recon(k,:,:)));
    photon_temp = squeeze(photons(k,:,:));
    star1_r(k) = round(sum(sum(temp(:,1:128))));
    star2_r(k) = round(sum(sum(temp(:,129:256))));
    star1(k) = round(sum(sum(photon_temp(:,1:10))));
    star2(k) = round(sum(sum(photon_temp(:,11:20))));
    figure(2), subplot(111),
    imagesc(temp), axis([119 138 119 138]), axis square, axis xy
    colorbar('vert'), title(['bin = ' num2str(k)]), pause(.01)
end

figure(1)
subplot(211), plot(bin_low_cent_high(:,2), star1,'r+',bin_low_cent_high(:,2), star1_r,'bo'),
legend('Original Star 1 @ T = 10000K', 'Reconstructed Star 1')
xlabel('Wavelength - \mum'), ylabel('Number of Photons')
```

```
subplot(212), plot(bin_low_cent_high(:,2), star2,'rx',bin_low_cent_high(:,2), star2_r,'gd')
legend('Original Star 2 @ T = 5000K','Reconstructed Star 2')
xlabel('Wavelength - \mum'), ylabel('Number of Photons')

C = [star1_r' star1' star1_r'./star1'  star2_r' star2' star2_r'./star2']

whos
toc
```

C.4.0.3 Recon_vector_big_scene. This file shows an example of the implementation of the vector reconstruction algorithm.

```
clear all
tic

load bin_data_15

%load('...Final Code\Monochromatic Bars\at_dectr_vectors_slab1s_15')
%load('...Final Code\Monochromatic Bars\at_dectr_vectors_slab2s_15')
%load('...Final Code\Monochromatic Bars\at_dectr_all_slab1s_15')
%load('...Final Code\Monochromatic Bars\at_dectr_all_slab1s_15_noisy')
%load('...Final Code\Monochromatic Bars\at_dectr_all_slab2s_15')
load('...Final Code\Monochromatic Bars\at_dectr_all_slab2s_15_noisy')
load OTF15vectors_1o5e

%load('...Final Code\Monochromatic Bars\at_dectr_vectors_slab1b_15')
%load('...Final Code\Monochromatic Bars\at_dectr_vectors_slab2b_15')
%load('...Final Code\Monochromatic Bars\at_dectr_all_slab1b_15')
%load('...Final Code\Monochromatic Bars\at_dectr_all_slab1b_15_noisy')
%load('...Final Code\Monochromatic Bars\at_dectr_all_slab2b_15')
%load('...Final Code\Monochromatic Bars\at_dectr_all_slab2b_15_noisy')
%load OTF15vectors_1e

rot_angle = length(bin_low_cent_high);
DS = 256;

data_cube = at_dectr_x;
%data_cube = at_dectr_y;

recon = ones(rot_angle,DS);

for iters = 1:1000
    est_data1 = zeros(rot_angle,DS);
    for angle = 1:rot_angle;
        temp6 = zeros(1,DS);
        for shift = 1:rot_angle;
            temp4 = fftshift(fft(recon(shift,:)));
            temp5 = temp4 .* conj(squeeze(otf_dectrA_x(shift,angle,:))');
%            temp5 = temp4 .* conj(squeeze(otf_dectrA_y(shift,angle,:))');
            temp6 = real(ifft(ifftshift(temp5))) + temp6;
        end
        est_data1(angle,:) = ifftshift(temp6);
    end
    differ = sum(sum(abs(est_data1-data_cube)))
    MSE(iters) = differ;
    map = (est_data1==0);
    est_data1 = (1-map) .* data_cube ./ (est_data1 + map) + map;

    for shift = 1:rot_angle;
        temp1 = zeros(1,DS);
        for angle = 1:rot_angle;
            temp2 = fftshift(fft(est_data1(angle,:)));
            temp3 = temp2 .* ((squeeze(otf_dectrB_x(shift,angle,:))'));
%            temp3 = temp2 .* ((squeeze(otf_dectrB_y(shift,angle,:))'));
            temp1 = real(ifft(ifftshift(temp3))) + temp1;
        end
        est_data2(shift,:) = ifftshift(temp1);
    end

    recon = recon .* est_data2 / rot_angle;
    count = iters

end

slab_r = round(sum(recon'));
slab = round(sum(photons_x'));
C = [slab_r; slab; slab_r-slab]'

 h = 1;
for k = [1 4 8 12 15]
    figure(3), subplot(3,2,h), temp = photons_x(k,:);
    plot(temp), axis([1 20 0 max(temp)+2]) ,
    title(['Original Vector @ \lambda = ' num2str(bin_low_cent_high(k,2)) ' \mum'])
    ylabel('Number of Photons')
    figure(2), subplot(3,2,h), temp1 = recon(k,:);
    plot(temp1),  axis([119 138 0 max(temp1)+2]),
    title(['Reconstructed Vector @ \lambda = ' num2str(bin_low_cent_high(k,2)) ' \mum'])
    ylabel('Number of Photons')
    h = h + 1;
end

figure(1)
subplot(111), plot(bin_low_cent_high(:,2), slab,'r+',bin_low_cent_high(:,2), slab_r,'bo'),
legend('Original Bar', 'Reconstructed Bar')
xlabel('Wavelength - \mum'), ylabel('Number of Photons')

%whos
toc
```

C.4.0.4 Recon_vector_15_atm_combo. This file shows an example of the implementation of the vector atmospheric reconstruction algorithms.

```
clear all;
tic

%load('...Final Code\Fireball 1\at_dectr_fball4K_15_atm'),
%load('...Final Code\Fireball 1\at_dectr_fball10K_15_atm'),
%load('...Final Code\Fireball 1\at_dectr_fball16K_15_atm'),
%load('...Final Code\Fireball Rings\at_dectr_fball_rings1_15_atm'),
load('...Final Code\Fireball Spots\at_dectr_fball_spots1_15_atm')

load bin_data_15
load OTF15vectors_1e

load Atmo_data_15
atmo = atmo_av_trans; T = sum(atmo);

rot_angle = length(atmo_av_trans);
DS = 256;

data_cube = at_dectr_x;
%data_cube = at_dectr_y;

recon = ones(rot_angle,DS);
recone = ones(rot_angle,DS);

for iters = 1:1000
    est_data1 = zeros(rot_angle,DS);
    est_data1e = zeros(rot_angle,DS);
    for angle = 1:rot_angle;
        temp6 = zeros(1,DS);
        temp6e = zeros(1,DS);
        for shift = 1:rot_angle;
            temp4 = fftshift(fft(recon(shift,:)));
            temp5 = temp4 .* conj(squeeze(otf_dectrA_x(shift,angle,:))');
%            temp5 = temp4 .* conj(squeeze(otf_dectrA_y(shift,angle,:))');
            temp6 = real(ifft(ifftshift(temp5))) + temp6;

            temp4e = fftshift(fft(recone(shift,:)));
            temp5e = temp4e .* conj(squeeze(otf_dectrA_x(shift,angle,:))') * atmo(shift) ;
%            temp5e = temp4e .* conj(squeeze(otf_dectrA_y(shift,angle,:))');
            temp6e = real(ifft(ifftshift(temp5e)))  + temp6e;
        end
        est_data1(angle,:) = ifftshift(temp6);
        est_data1e(angle,:) = ifftshift(temp6e);
    end
    differ = [sum(sum(abs(est_data1-data_cube))) sum(sum(abs(est_data1e-data_cube)))]
    MSE(iters,:) = differ;
    map = (est_data1==0);
    est_data1 = (1-map) .* data_cube ./ (est_data1 + map) + map;
    map = (est_data1e==0);
    est_data1e = (1-map) .* data_cube ./ (est_data1e + map) + map;

    for shift = 1:rot_angle;
        temp1 = zeros(1,DS);
        temp1e = zeros(1,DS);
        for angle = 1:rot_angle;
            temp2 = fftshift(fft(est_data1(angle,:)));
            temp3 = temp2 .* ((squeeze(otf_dectrB_x(shift,angle,:))'));
%            temp3 = temp2 .* ((squeeze(otf_dectrB_y(shift,angle,:))'));
            temp1 = real(ifft(ifftshift(temp3))) + temp1;

            temp2e = fftshift(fft(est_data1e(angle,:)));
            temp3e = temp2e .* ((squeeze(otf_dectrB_x(shift,angle,:))')) * atmo(angle);
%            temp3e = temp2e .* ((squeeze(otf_dectrB_y(shift,angle,:))'));
            temp1e = real(ifft(ifftshift(temp3e))) + temp1e;
        end
        est_data2(shift,:) = ifftshift(temp1);
        est_data2e(shift,:) = ifftshift(temp1e);
    end

    recon = recon .* est_data2 / rot_angle;
    recone = recone .* est_data2e / T;
    count = iters

end

E = round(sum(recon')');
for k = 1:rot_angle
    recon(k,:,:) = recon(k,:,:) / atmo_av_trans(k);
%    photons_x(k,:) = photons_x(k,:) / atmo_av_trans(k);
%    photons_y(k,:) = photons_y(k,:) / atmo_av_trans(k);
end
A = round(sum(recon')');
B =  sum(photons_x')';
D = round(sum(recone')');
C = [B A A./B D D./B]

figure(1)
subplot(311), plot(bin_low_cent_high(:,2),B','r+',bin_low_cent_high(:,2),E','bo')
subplot(312), plot(bin_low_cent_high(:,2),B','r+',bin_low_cent_high(:,2),A','bo')
subplot(313), plot(bin_low_cent_high(:,2),B','r+',bin_low_cent_high(:,2),D','bo')
%whos
toc
```

C.5 Miscellaneous CTIS Matlab® Files

C.5.0.1 Index_inter_3. This file interpolates the spectral refactive indices of the DVP optical materials.

```
function [Wave_Index] = Index_inter_3(bin_cent)
% Returns interpolated refractive indices for "bins"
% number of wavelength bins between 2 to 5 microns
%clear all

%load bin_data_15
%bin_cent = bin_low_cent_high(:,2);

lam3 = 2:0.5:5;
n3 = [1.37875 1.37327 1.3666 1.35868 1.34942 1.33875 1.32661];

lam4 = [1.97009 2.1526 2.32542 2.5766 2.6738 3.2434 3.422 5.138];
n4 = [1.46470 1.46412 1.46356 1.46271 1.46237 1.46017 1.45941 1.45014];

grad3 = diff(n3) / 0.5;
grad4 = diff(n4) ./ diff(lam4);

b3 = n3(1:6) - grad3 .* lam3(1:6);
b4 = n4(1:7) - grad4 .* lam4(1:7);

for i = 1 :length(bin_cent)
    j = 1;
    check = 0;
    while check == 0;
        if (bin_cent(i) >= lam3(j) & bin_cent(i) < lam3(j+1))
            n3_lamda(i) = grad3(j) * bin_cent(i) + b3(j);
            check = 1;
        else
            j = j + 1;
        end
    end
end

for i = 1 :length(bin_cent)
    j = 1;
    check = 0;
    while check == 0;
        if (bin_cent(i) >= lam4(j) & bin_cent(i) < lam4(j+1))
            n4_lamda(i) = grad4(j) * bin_cent(i) + b4(j);
            check = 1;
        else
            j = j + 1;
        end
    end
end

Wave_Index = [bin_cent n3_lamda' n4_lamda'];

%figure(1), subplot(211), plot(lam3,n3,'r*',bin_cent,n3_lamda,'b.'), grid on
%axis([1.75 5.25 1.325 1.382])
%xlabel('Wavelength (\mum)'), ylabel('Refractive Index (n_2)')
%title('Spectral Refractive Index for LiF')
```

```
%legend('Spectral refractive index','Interpolated refractive index',3)
%subplot(212), plot(lam4,n4,'r*',bin_cent,n4_lamda,'b.'), grid on
%axis([1.75 5.25 1.45 1.465])
%xlabel('Wavelength (\mum)'), ylabel('Refractive Index (n_3)')
%title('Spectral Refractive Index for BaF_2')
%legend('Spectral refractive index','Interpolated refractive index',3)
%whos
```

C.5.0.2 Make_OTF_dectr_no_fftshift. This file calculates the 2D OTF of the modelled CTIS.

```
clear all tic

%load Shifts_15_1o5e
load Shifts_15_1e
load PSF_unshifted_15

rot_angle = 15;
DS = 256;

for angle = 1:rot_angle;
    m1 = m(:,angle);
    n1 = n(:,angle);
    for shift =  1 : rot_angle;
        mm1 = m1(shift)-10:m1(shift)+10;
        nn1 = n1(shift)-10:n1(shift)+10;
        psf_back = zeros(DS);
        psf_back(mm1,nn1) =  squeeze(PSF(shift,:,:)) + psf_back(mm1,nn1);
        otf_dectr(shift,:,:) = (fft2(psf_back));
    end
    save(['...Final Code\OTF15big no fftshift\otf_dectrA_' num2str(angle)],'otf_dectr')
    angle
end

for shift =  1:rot_angle;
    m1 = m(shift,:);
    n1 = n(shift,:);
    for angle =  1 : rot_angle;
        mm1 = m1(angle)-10:m1(angle)+10;
        nn1 = n1(angle)-10:n1(angle)+10;
        psf_back = zeros(DS);
        psf_back(mm1,nn1) =  squeeze(PSF(shift,:,:)) + psf_back(mm1,nn1);
        otf_dectr(angle,:,:) = (fft2(psf_back));
    end
    save(['...Final Code\OTF15big no fftshift\otf_dectrB_' num2str(shift)],'otf_dectr')
    shift
end

toc
```

C.5.0.3 Make_OTF_dectr_vector. This file calculates the 1D OTF of the modelled CTIS.

```
clear all
tic

load Shifts_15_1e, load PSF_unshifted_15
%load Shifts_15_1o5e, load PSF_unshifted_15
%load Shifts_30_1e, load PSF_unshifted_30
%load Shifts_50_1o5e, load PSF_unshifted_50
%load Shifts_50_1e, load PSF_unshifted_50

rot_angle = length(m);
DS = 256;

for angle = 1:rot_angle;
    m1 = m(:,angle);
    n1 = n(:,angle);
    for shift =  1 : rot_angle;
        mm1 = m1(shift)-10:m1(shift)+10;
        nn1 = n1(shift)-10:n1(shift)+10;
        psf_back = zeros(DS);
        psf_back(mm1,nn1) =  squeeze(PSF(shift,:,:)) + psf_back(mm1,nn1);
        temp = fftshift(fft2(psf_back));
        otf_dectrA_x(shift,angle,:) = temp(129,:);
        otf_dectrA_y(shift,angle,:) = temp(:,129);
    end
    angle
end

for shift =  1:rot_angle;
    m1 = m(shift,:);
    n1 = n(shift,:);
    for angle =  1 : rot_angle;
        mm1 = m1(angle)-10:m1(angle)+10;
        nn1 = n1(angle)-10:n1(angle)+10;
        psf_back = zeros(DS);
        psf_back(mm1,nn1) =  squeeze(PSF(shift,:,:)) + psf_back(mm1,nn1);
        temp = fftshift(fft2(psf_back));
        otf_dectrB_x(shift,angle,:) = temp(129,:);
        otf_dectrB_y(shift,angle,:) = temp(:,129);
    end
    shift
end

whos
toc
```

C.5.0.4 Prism_rays_3. This file calculates the radial spectral shifts produced by the DVP.

```
function Shifts = Prism_rays_3(bin_cent)

%bins = 16;
a1 = pi/6;
a3 = (23+57/60)*pi/180;
%a3 = (23+38.5/60)*pi/180;
%a1 = (30.03)*pi/180;
%a3 = (23.96)*pi/180;


W_N = Index_inter_3(bin_cent);
n1 = 1;
n3 = W_N(:,2);
n4 = W_N(:,3);

Theta3 = asin(1./(2*n3)) - a1;
Theta4 = asin(n3./n4.*sin(Theta3)) + a3;
Theta5 = asin(n4./n1.*sin(Theta4)) - a3;

r = -0.5 * tan(Theta5);

%[Theta3 Theta4 Theta5 W_N(:,1)]
Shifts = [W_N(:,1) r];
%figure(10), stem(Shifts(:,1), Shifts(:,2)*1e3)
%axis([2 5 -7 5])
%axis([3.5 3.7 -1e-3 1e-3])
%title('Radial Wavelength Shift using Geometric Optics for Prism')
%xlabel('Wavelength - \mum'), ylabel('Shift - mm')
%grid on, axis square
```

C.5.0.5 PSFs_RS_fn. This file calculates the unshifted spectral PSF for light at the center wavelength of each bin.

```
function [I_PDs] = PSFs_RS_fn(bin_cent)
%clear all
%load bin_data_15
%bin_cent = bin_low_cent_high(:,2)

%format long g
Lens_radius = 0.05;
f = 0.5;
%lamda = 5e-6;
%bins = 15;              % Number of wavelength bins

num_ap_samples = 11;     % Must be odd!!. Assumes num_samples is same in both x & y direction
ends = (num_ap_samples - 1) / 2;
M1=[-ends:ends];
[m1,n1]=meshgrid(M1,M1);

num_det_samples = 21;     % Must be odd!!. Assumes num_samples is same in both x & y direction
ends2 = (num_det_samples - 1) / 2;
M2=[-ends2:ends2];
```

```
% Sample sizes in lens and detector
d_x_ap = (Lens_radius*2) / num_ap_samples  ;
d_y_ap = d_x_ap ;
d_x_dt = 1.22e-6/(Lens_radius*2) * .25;   %3.0500e-006;
%d_x_dt = 24e-6;
d_y_dt = d_x_dt;

% Calculation of point distances from lens to detector
count = 1;
for m2 = M2
    for n2=M2
%       R_Ds1(count,:,:) = sqrt((m1*d_x_ap - m2*d_x_dt).^2 + (n1*d_y_ap - n2*d_y_dt).^2 + f^2);
       R_Ds1(count,:,:)= f + (m1*d_x_ap - m2*d_x_dt).^2/(2*f) + (n1*d_y_ap - n2*d_y_dt).^2/(2*f);
       R_Ds2(count,:,:)= f^2 + (m1*d_x_ap - m2*d_x_dt).^2 + (n1*d_y_ap - n2*d_y_dt).^2;
           count = count +1;
    end
end

% Circular shape of lens
Lens_shape = ((m1*d_x_ap).^2+(n1*d_y_ap).^2) < Lens_radius^2;
% Divides 2 - 5 mircon band into "bins" number of wavelengths
%lamda = index_inter(bins-1); lamda = lamda(:,1) * 1e-6;
lamda = bin_cent * 1e-6;

for bigloop = 1 : length(bin_cent)
    Lens_transform = exp(-j*pi*((m1*d_x_ap).^2+(n1*d_y_ap).^2)/(lamda(bigloop)*f));
    U_out_lens = ones(num_ap_samples) .* Lens_shape .* Lens_transform;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Propagation from lens to detector
    count = 1;
    for m2 = M2
        for n2=M2
            r_ds1 = squeeze(R_Ds1(count,:,:));
            r_ds2 = squeeze(R_Ds2(count,:,:));
            count = count +1;
            U_PDs(m2+ends2+1,n2+ends2+1)=sum(sum((1./r_ds2 .* U_out_lens.* ...
                exp(j*2*pi*r_ds1./lamda(bigloop))) * ...
                sinc(m2*d_x_dt*d_x_ap*pi/(lamda(bigloop)*f))*...
                sinc(n2*d_y_dt*d_x_ap*pi/(lamda(bigloop)*f))*d_x_ap*d_y_ap));
        end
    end
    U_PDs=U_PDs*f/(j*lamda(bigloop));
    %U_PDs=U_PDs/(d_x_ap*j);
    I_PDs(bigloop,:,:) = U_PDs.*conj(U_PDs)*d_x_dt;
end

% Normalises PSFs to each have sum of one
for count = 1 : length(bin_cent)
    temp = squeeze(I_PDs(count,:,:));
    I_PDs(count,:,:) =  temp / sum(sum(temp));
end
```

C.5.0.6 Pixel_temperature_fit. This file estimates the reconstructed temperatures of the binary star test cases.

```
clear all, tic
% Atmospheric coefficient data
load('...Final Code\Atmo_data_15')
% Bin data
load('...Final Code\bin_data_15'),
bins = length(bin_low_cent_high);
int = (5-2)/bins;
lamda =2:int:5-int;
% Test data
%load('...Final Code\Binary Star\data_run_2D_bin_star_15_no_atm_100iters')
%load('...Final Code\Binary Star\data_run_2D_bin_star_15_atm_100iters')
%load('...Final Code\Binary Star\data_run_bin_star_15_no_atm_1000iters')
load('...Final Code\BinaryStar\data_run_bin_star_15_atm_1000iters')

% Original photons
%star1 = photons(:,11,7);
%star2 = photons(:,11,14);

% Reconstructed photons 2D
%star1_r = recon(:,129,125);           % Just looks at individual pixel locations for stars
%star2_r = recon(:,129,132);
%star1_r = sum(sum(recon(:,:,1:128),2),3); % Sums photons for all pixels for relevant stars 2D
%star2_r = sum(sum(recon(:,:,129:256),2),3);

% Reconstructed photons 1D
star1_r = recon(:,125);                % Just looks at individual pixel locations for stars
star2_r = recon(:,132);
%star1_r = sum(recon(:,1:128),2);      % Sums photons for all pixels for relevant stars 1D
%star2_r = sum(recon(:,129:256),2);

% Fundamental constants.
c = 299792458;
light_year = c * 60 * 60 * 24 * 365; % units of meters
h = 6.6260755e-34;
kB = 1.380658e-23;

% Star Parameters
sol_rad =  6.96e10; % Solar radii in units of centimeters % 1 solar radii = 6.96e8 meters
R1 = 2.5 ;          % Radius of star 1 in units of Solar Radii;
R2 = 1.1 ;          % Radius of star 2 in units of Solar Radii;
A1 = pi*(R1*sol_rad)^2;  % Projected area of star1 in centimeters
A2 = pi*(R2*sol_rad)^2;  % Projected area of star2 in centimeters
Dist = 50; % Light years

% Forward Optics Parameters
D = .1; % diameter of the telescope in meters;
SR1 = (D/2)^2*pi/(Dist * light_year)^2; % Solid angle subtended by star 1 to sensor
SR2 = (D/2)^2*pi/(Dist * light_year)^2; % Solid angle subtended by star 2 to sensor

% Detector Parameters
dt=.001; % Integration time in seconds of the CCD camera
```

```
% Results for summed pixels
%star1_temp_guess = 10022;       % 2D data with no Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 4873;
%star1_temp_guess = 8324;        % 2D data with no Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 3948;        %  Using single, peak pixel for each star from reconstruction
%star1_temp_guess = 9964;        % 2D data with Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 4804;                   %
%star1_temp_guess = 10015;       % 2D data with Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 4909;        % Removal of low atmo coeff bins, ie, bin 4 and 12

%star1_temp_guess = 10015;       % 1D data with no Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 4920;
%star1_temp_guess = 9590;        % 1D data with no Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 4527;        %  Using single, peak pixel for each star from reconstruction
%star1_temp_guess = 10070;       % 1D data with Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 4666;                   %
%star1_temp_guess = 10010;       % 1D data with Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 4959;        % Removal of low atmo coeff bins, ie, bin 4 and 12

% Results for single peak pixel
%star1_temp_guess = 8324;        % 2D data with no Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 3948;
%star1_temp_guess = 8322;        % 2D data with Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 3953;
%star1_temp_guess = 8473;        % 2D data with  Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 4095;        % Removal of low atmo coeff bins, ie, bin 4 and 12

%star1_temp_guess = 9590;        % 1D data with no Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 4527;
%star1_temp_guess = 9499;        % 1D data with Atm : final resolution 1 degree Kelvin
%star2_temp_guess = 4369;
star1_temp_guess = 9624;         % 1D data with  Atm : final resolution 1 degree Kelvin
star2_temp_guess = 4642;         % Removal of low atmo coeff bins, ie, bin 4 and 12

big_count = 1;
limit = 10;
interval = limit / 10
temp_range = -limit : interval : limit;
% Mask to remove low transmission bins from calculation
%mask = ones(15,1);
mask = [1 1 1 0 1 1 1 1 1 1 1 0 1 1 1]’;

for delta_temp =  temp_range % start big for loop

    truth_1 = star1_temp_guess + delta_temp;   % Star 1 has a temperature of 5000 K
    truth_2 = star2_temp_guess + delta_temp;   % Star 2 has a temperature of 10000 K
    truth = [truth_1 truth_2];
    % Gives the spectral photon radiance [photons/sec - cm^2 - sr - um] as a
    % sum of "sub_lamda" sub-bin areas in each wavelength bin.
    sub_lamda = 100;
    for index = 1:bins
        lamda_base = lamda(index):int/sub_lamda:lamda(index) + int - int/sub_lamda;
        temporary = 0;
        for count = 1:sub_lamda
            temporary = temporary + (2 * c * 1e14) ./ (lamda_base(count)^4 .* ...
                (exp(h*c./(lamda_base(count)*1e-6*kB.*truth)) - 1)) * int/sub_lamda;
        end
        rad_p(index,:) = temporary;
    end

    % Calculates photons at detector over integration time and solid angle
    photons_1 = rad_p(:,1) * dt*A1*SR1;
    photons_2 = rad_p(:,2) * dt*A2*SR2;

    % Calculates error between reconstructed photons and photons produced from temperature guess
    Error_Star1(big_count) = (sum(mask.*(photons_1 - star1_r).^2))/15;
    Error_Star2(big_count) = (sum(mask.*(photons_2 - star2_r).^2))/15;
    big_count = big_count+1;
end  % end big for loop

% Calculates the minimum error, ie, best temperature fit to reconstructed photons
[aaa,bbb] = min(Error_Star1); [ccc,ddd] = min(Error_Star2);
Best_fit_star_1 = star1_temp_guess + temp_range(bbb)
Best_fit_star_2 = star2_temp_guess + temp_range(ddd)

figure(3),
subplot(211), plot(temp_range,Error_Star1), grid on, title('Error in Star 1')
subplot(212), plot(temp_range,Error_Star2), grid on, title('Error in Star 2')
toc
```

C.5.0.7 Pixel_temperature_fit_2D_fball. This file estimates the reconstructed temperatures in the 2D fireball test cases.

```
clear all, tic
% Atmospheric coefficient data
load('...Final Code\Atmo_data_15')
% Bin data
load('...Final Code\bin_data_15'),
bins = length(bin_low_cent_high);
int = (5-2)/bins;
lamda = 2:int:5-int;
% Test data: Concentric Rings Fireball
load('...Fireball Rings\data_run_2D_fball_rings1_15_100iters_atm'),
load('...Final Code\Fireball Rings\photons_fball_rings1_15')
reconAE = recon;
load('...Final Code\Fireball Rings\data_run_2D_fball_rings1_15_100iters')

% Test data: Hot-Spots Fireball
%load('...Final Code\Fireball Spots\data_run_2D_fball_spots1_15_100iters_atm')
%reconAE = recon;
%load('...Final Code\Fireball Spots\data_run_2D_fball_spots1_15_100iters')

% Reconstructed photons 2D: Concentric Rings Fireball
pix_loc = 145;
pixel1_r = recon(:,pix_loc,pix_loc);   % Just looks at a single pixel location of fireball.
pixel1_rAE = reconAE(:,pix_loc,pix_loc);

% Reconstructed photons 2D: Hot-Spots Fireball
%pix_loc = [129 129];            % Fireball center 600K
%pix_loc = [119 119];            % Fireball 1000K
%pix_loc = [119 139];            % Fireball 1500K
%pix_loc = [139 119];            % Fireball 1200K
```

```
%pix_loc = [139 139];            % Fireball 1600K
%pix_loc = [160 160];            % Background 300K
%pixel1_r = recon(:,pix_loc(1),pix_loc(2));  % Just looks at a single pixel location of fireball.
%pixel1_rAE = reconAE(:,pix_loc(1),pix_loc(2));

% Fundamental constants.
c = 299792458;
h = 6.6260755e-34;
kB = 1.380658e-23;

% Fireball Parameters
%Projected area of a square pixel at 3258m entered in metres^2 converted to centimetres^2
A = 3.258^2 * 1000;
Dist = 3258; % Distance for fireball to sensor in meters

% Forward Optics Parameters
D = .1; % diameter of the telescope in meters;
SR = D*D*pi/(4*Dist); % Solid angle subtended by sensor to fireball range

% Detector Parameters
dt=.001; % Integration time in seconds of the CCD camera

% Reconstructed temperatures for Concentric Rings fireball
%ball_temp_guess = 1605;  % Concentric rings fireball (1600K) @ pixel location (129,129)   1602
%ball_temp_guess = 1499;  % Concentric rings fireball (1500K) @ pixel location (135,135)   1499
%ball_temp_guess = 1200;  % Concentric rings fireball (1200K) @ pixel location (138,138)   1203
%ball_temp_guess = 983;   % Concentric rings fireball (1000K) @ pixel location (142,142)    981
ball_temp_guess = 823;    % Concentric rings fireball  (800K) @ pixel location  (145,145)   819
%ball_temp_guess = 296;   % Concentric rings fireball  (300K) @ pixel location  (160,160)   299

% Reconstructed temperatures for Hot-Spots fireball
%ball_temp_guess = 609;   % Hot-Spots fireball (600K)  @ pixel location (129,129)  571
%ball_temp_guess = 994;   % Hot-Spots fireball (1000K) @ pixel location (119,119)  994
%ball_temp_guess = 1475;  % Hot-Spots fireball (1500K) @ pixel location (129,139) 1476
%ball_temp_guess = 1179;  % Hot-Spots fireball (1200K) @ pixel location (139,119) 1176
%ball_temp_guess = 1657;  % Hot-Spots fireball (1600K) @ pixel location (139,139) 1665
%ball_temp_guess = 292;   % Hot-Spots fireball (300K)  @ pixel location (160,160)  299

big_count = 1;
limit = 10;
interval = limit / 10
temp_range = -limit : interval : limit;
% Mask to remove low transmission bins from calculation
mask = [1 1 1 0 1 1 1 1 1 1 1 0 1 1 1]';

for delta_temp =  temp_range % start big for loop

    truth = (ball_temp_guess + delta_temp); %units of degrees kelvin

    % Gives the spectral photon radiance [photons/sec - cm^2 - sr - um] as a
    % sum of "sub_lamda" sub-bin areas in each wavelength bin.
    sub_lamda = 100;
    for index = 1:bins
        lamda_base = lamda(index):int/sub_lamda:lamda(index) + int - int/sub_lamda;
        temporary = 0;
        for count = 1:sub_lamda
```

```
            temporary = temporary + (2 * c * 1e14) ./ (lamda_base(count)^4 .* ...
                (exp(h*c./(lamda_base(count)*1e-6*kB.*truth)) - 1)) * int/sub_lamda;
        end
        rad_p(index,:) = temporary;
    end

    % Calculates photons at detector over integration time and solid angle
    photons_1 = rad_p * dt*A*SR;

    % Calculates error between reconstructed photons and photons produced from temperature guess
    Error_Clean(big_count) = (sum(mask.*(photons_1 - pixel1_r).^2))/15;
    Error_Atmo_Est(big_count) = (sum((photons_1 - pixel1_rAE ).^2 ))/15;
       % Compensating low Atm coefficient bins by removal
    Error_Atmo_Est_comp(big_count) = (sum(mask.*(photons_1 - pixel1_rAE).^2))/15;
    big_count = big_count+1;
end  % end big for loop

% Calculates the minimum error, ie, best temperature fit to reconstructed photons
[aaa,bbb] = min(Error_Clean);
[eee,fff] = min(Error_Atmo_Est);
[iii,jjj] = min(Error_Atmo_Est_comp);
Best_fit_clean = ball_temp_guess + temp_range(bbb)
Best_fit_Atmo_Est = ball_temp_guess + temp_range(fff)
Best_fit_Atmo_Est_comp = ball_temp_guess + temp_range(jjj)

figure(3),
subplot(311), plot(temp_range,Error_Clean), grid on
subplot(312), plot(temp_range,Error_Atmo_Est), grid on
subplot(313), plot(temp_range,Error_Atmo_Est_comp), grid on
toc
```

C.5.0.8 Pixel_temperature_fit_uni_fball. This file estimates the reconstructed temperatures in the uniform fireball test cases.

```
clear all, tic
% Atmospheric coefficient data
load('...Final Code\Atmo_data_15')
% Bin data
load('...Final Code\bin_data_15'),
bins = length(bin_low_cent_high);
int = (5-2)/bins;
lamda = 2:int:5-int;
% Test data: Static Fireballs
%load('...Final Code\Fireball 1\data_run_1D_uni_fball4K_1000iters_atm')
%reconAI = recon;
%reconAE = recone;
%load('...Final Code\Fireball 1\data_run_1D_uni_fball4K_1000iters')
%load('...Final Code\Fireball 1\data_run_1D_uni_fball10K_1000iters_atm')
%reconAI = recon;
%reconAE = recone;
%load('...Final Code\Fireball 1\data_run_1D_uni_fball10K_1000iters')
load('...Final Code\Fireball 1\data_run_1D_uni_fball16K_1000iters_atm')
reconAI = recon;
reconAE = recone;
%load('...Fireball 1\data_run_1D_uni_fball16K_1000iters')
```

```
% Test data: Evolving fireballs
load('...Final Code\Fireball Evolve\data_run_1D_fball16K_10K_4K_1000iters')
%load('...Final Code\Fireball Evolve\data_run_1D_fball16K_14K_12K_1000iters')
%load('...Final Code\Fireball Evolve\data_run_1D_fball16K_13K_1000iters')
%load('...Final Code\Fireball Evolve\data_run_1D_fball16K_15K_1000iters')

% Reconstructed photons 1D
pix_loc = 127;
pixel1_r = recon(:,pix_loc) / 10;       % Just looks at center pixel location of fireball.
pixel1_rAI = reconAI(:,pix_loc) / 10;   % Divides by 10 as there are 10 pixels across the
pixel1_rAE = reconAE(:,pix_loc) / 10;   % diameter of the fireball.

% Fundamental constants.
c = 299792458;
light_year = c * 60 * 60 * 24 * 365; % units of meters
h = 6.6260755e-34;
kB = 1.380658e-23;

% Fireball Parameters
%Projected area of a square pixel at 3258m entered in metres^2 converted to centimetres^2
A = 3.258^2 * 1000;
Dist = 3258; % Distance for fireball to sensor in meters

% Forward Optics Parameters
D = .1; % diameter of the telescope in meters;
SR = D*D*pi/(4*Dist); % Solid angle subtended by sensor to fireball range

% Detector Parameters
dt=.001; % Integration time in seconds of the CCD camera

% Guesses for static fireballs
%ball_temp_guess = 1574;   % 1600K fireball @ pixel location 129
%ball_temp_guess = 1652;   % 1600K fireball @ pixel location 127  1566  1652
%ball_temp_guess = 987;    % 1000K fireball @ pixel location 127  1234   987
%ball_temp_guess = 424;    %  400K fireball @ pixel location 127   904
%ball_temp_guess = 411;    %  400K fireball @ pixel location 125
% Guesses for evolving fireballs
%ball_temp_guess = 1515;   % 1600 to 1500K fireball  @ pixel location 128
%ball_temp_guess = 1436;   % 1600 to 1300K fireball  @ pixel location 127
%ball_temp_guess = 1402;   % 1600 to 1400 to 1200K fireball @ pixel location 127
ball_temp_guess = 1233;    % 1600 to 1000 to 400K fireball  @ pixel location 127

big_count = 1;
limit = 10;
interval = limit / 10
temp_range = -limit : interval : limit;
mask = ones(15,1);
% Mask to remove low transmission bins from calculation
%mask = [1 1 1 0 1 1 1 1 1 1 1 0 1 1 1]';

for delta_temp =  temp_range % start big for loop

    truth = (ball_temp_guess + delta_temp); %units of degrees kelvin
    % Gives the spectral photon radiance [photons/sec - cm^2 - sr - um] as a
    % sum of "sub_lamda" sub-bin areas in each wavelength bin.
    sub_lamda = 100;
    for index = 1:bins
        lamda_base = lamda(index):int/sub_lamda:lamda(index) + int - int/sub_lamda;
        temporary = 0;
        for count = 1:sub_lamda
            temporary = temporary + (2 * c * 1e14) ./ (lamda_base(count)^4 .* ...
                (exp(h*c./(lamda_base(count)*1e-6*kB.*truth)) - 1)) * int/sub_lamda;
        end
        rad_p(index,:) = temporary;
    end

    % Calculates photons at detector over integration time and solid angle
    photons_1 = rad_p * dt*A*SR;

    % Calculates error between reconstructed photons and photons produced from temperature guess
    Error_Clean(big_count) = sum((photons_1 - pixel1_r).^2)/15;
    Error_Atmo_Inv(big_count) = sum((photons_1 - pixel1_rAI).^2)/15;
    % Compensating low Atm coefficient bins by removal
    Error_Atmo_Inv_comp(big_count) = sum(mask.*(photons_1 - pixel1_rAI).^2)/15;
    Error_Atmo_Est(big_count) = sum((photons_1 - pixel1_rAE ).^2 )/15;
    % Compensating low Atm coefficient bins by removal  :)
    Error_Atmo_Est_comp(big_count) = sum(mask.*(photons_1 - pixel1_rAE).^2)/15;
    big_count = big_count+1;
end  % end big for loop

% Calculates the minimum error, ie, best temperature fit to reconstructed photons
[aaa,bbb] = min(Error_Clean);
[ccc,ddd] = min(Error_Atmo_Inv);
[eee,fff] = min(Error_Atmo_Est);
[ggg,hhh] = min(Error_Atmo_Inv_comp);
[iii,jjj] = min(Error_Atmo_Est_comp);
Best_fit_clean = ball_temp_guess + temp_range(bbb)
Best_fit_Atmo_Inv = ball_temp_guess + temp_range(ddd)
Best_fit_Atmo_Est = ball_temp_guess + temp_range(fff)
Best_fit_Atmo_Inv_comp = ball_temp_guess + temp_range(hhh)
Best_fit_Atmo_Est_comp = ball_temp_guess + temp_range(jjj)

figure(3),
subplot(511), plot(temp_range,Error_Clean), grid on
subplot(512), plot(temp_range,Error_Atmo_Inv), grid on
subplot(513), plot(temp_range,Error_Atmo_Est), grid on
subplot(514), plot(temp_range,Error_Atmo_Inv_comp), grid on
subplot(515), plot(temp_range,Error_Atmo_Est_comp), grid on
toc
```

Bibliography

1. Aikio, Mauri. "Hyperspectral Prism-Grating-Prism Imaging Spectrograph." University of Oulu PhD Dissertation, June 2001.

2. Brodzik, Andrzej K. and Jonathan M. Mooney. "Convex Projections Algorithm for Restoration of Limited-angle chromotomographic images," *Journal of the Optical Society of America (JOSA)*, *16*(2):246–257 (1999).

3. Brodzik, Andrzej K. and Jonathan M. Mooney and Myoung An. "Image Restoration by Convex Projections: Application to Image Spectrometry," *The International Society of Optical Engineers (SPIE)*, *2819*:231–240 (1996).

4. Cain, Stephen. "OENG644 Class Notes." AFIT, 2004.

5. Col, Jeananda. "Enchanted Learning Website - Star Types." http://www.enchantedlearning.com/subjects/astronomy/stars/startypes.shtml, 1999.

6. Dearinger, Anthony J. "Simulating a Chromotomographic Sensor for Hyperspectral Imaging in the Infrared." AFIT Masters Thesis, 12 March 2004.

7. Dereniak, Eustace L. and Glenn D. Boreman. *Infrared Detectors and Systems*. New York, NY: John Wiley and Sons, Inc, 1996.

8. Dills, Anthony N. and Glen P. Perram. *Brilliant Flash II Preliminary Test Report: AFIT FTIR and Imaging Sensors*. Technical Report, AFIT, 30 July 2003.

9. Gat, Nahum and Suresh Subramanian. "Spectral Imaging: Technology and Applications," *Opto-Knowledge Systems, Inc: Hyperspectrum News Letter*, *3*(1):1–30 (February 1997).

10. Goodman, Joseph W. *Introduction to Fourier Optics, 2nd edition*. Stanford University: McGraw-Hill, 1996.

11. Gustke, Kevin C. "Reconstruction Algorthim Characterization and Performance Monitoring in Limited-Angle Chromotomography." AFIT Masters Thesis, 15 March 2004.

12. Hecht, Eugene. *Optics, 4th edition*. San Francisco, CA: Addison-Wesly, 2002.

13. LeMaster, Daniel A. "Design and Model Verification of an Infrared Chromotomographic Imaging System." AFIT Masters Thesis, November 2004.

14. Lord, Steven D. *A New Software Tool for Computing Earth's Atmospheric Transmission in the Near and Far Infrared Radiation*. Technical Report NASA Tech. Memo. 103957, Gemini Observatory, 1992.

15. Mooney, Jonathan M. "Angularly Multiplexed Spectral Imager," *The International Society of Optical Engineers (SPIE)*, *2480*(65):65–77 (1995).

16. Mooney, Jonathan M. and Andrzej K. Brodzik and Myoung An. "Principal Component Analysis in Limited-angle Chromotomography," *The International Society of Optical Engineers (SPIE)*, *3118*:170–178 (1997).

17. Papoulis, Athanasios and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes, 4th edition*. New York, NY: McGraw-Hill Higher Education, 2002.

18. Strang, Gilbert. *Linear Algebra and its Applications, 3rd edition*. Orlando, FL: Harcourt College Publishers, 1988.

19. Surface Optics Corporation. *Hyperspectral Applications*. San Diego, CA: http://www.surfaceoptics.com/Products/Hyperspectral/HyperspectralApps.htm, 2001.

20. Swokowski, Earl W. *Calculus, 5th edition*. Boston, MA: PWS-Kent Publishing Company, 1991.

21. Van Trees, Harry L. *Detection, Estimation and Modulation Theory, Part 1*. New York, NY: John Wiley and Sons, Inc, 2001.

22. Wolfe, William L. and George J. Zissis. *The Infrared Handbook, revised edition, 4th printing*. Environmental Research Institute of Michigan: IRIA Series in Infrared and Electro-Optics, 1993.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 21 03 2005	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From - To)* Jun 2004 - Mar 2005

4. TITLE AND SUBTITLE

RECONSTRUCTION OF CHROMOTOMOGRAPHIC IMAGING SYSTEM INFRARED HYPERSPECTRAL SCENES

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Gould, Malcolm G, Flight Lieutenant, Royal Australian Air Force

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 641
Wright-Patterson AFB OH 45433-5648

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/05-04

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr. Michael Eismann, DR-IV
Electro-Optical Technology Division, EO Threat and Target Detection
Air Force Research Laboratory,
AFRL/SNJT
2241 Avionics Circle
Wright-Patterson AFB OH 45433

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Hyperspectral imagery providing both spatial and spectral information has diverse applications in remote sensing and scientific imaging scenarios. The development of the Chromotomographic Imaging System (CTIS) allows simultaneous collection of both spatial and spectral data by a two-dimensional (2D) focal plane detector array. Post-processing of the 2D detector data reconstructs the three-dimensional (3D) hyperspectral content of the imaged scene. This thesis develops Estimation Theory based algorithms for reconstructing the hyperspectral scene data. The initial algorithm developed reconstructs the 3D hyperspectral scene data cube. An additional algorithm reconstructs a matrix comprised of one spectral dimension and one compound spatial dimension. Methods for including the effects of atmospheric attenuation on the light over the propagation path are also included. The algorithms are evaluated using test cases consisting of blackbody point sources, monochromatic extended sources and blackbody extended sources. The results show good performance for reconstructing the absolute radiometry and spatial features of a hyperspectral scene data cube. These algorithms also do not significantly degrade in the presence of noisy detector data. The vector algorithm also exhibits stable performance behaviour when reconstructing a temporally evolving hyperspectral scene.

15. SUBJECT TERMS

Chromotomography, Hyperspectral Imagery, Spectrometry, Infrared, Image Reconstruction

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Stephen Cain (ENG), Stephen.Cain@afit.edu
U	U	U	U	208	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255 6565 ext:4408